



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY**

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

NÁVRH OPTIMALIZACE DÍLČÍ ČÁSTI INFORMAČNÍHO SYSTÉMU

THE PROPOSAL FOR AN OPTIMIZATION OF THE INFORMATION SYSTEM SUBSEGMENT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

PETR ČMOK

VEDOUcí PRÁCE

SUPERVISOR

Ing. BERNARD NEUWIRTH, Ph.D.

BRNO 2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Čmok Petr

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

Návrh optimalizace dílčí části informačního systému

v anglickém jazyce:

The Proposal for an Optimization of the Information System Subsegment

Pokyny pro vypracování:

Úvod

Cíle práce, metody a postupy zpracování

Teoretická východiska práce

Analýza současného stavu

Vlastní návrhy řešení

Závěr

Seznam použité literatury

Přílohy

Seznam odborné literatury:

BASL, Josef a Roman BLAŽÍČEK. Podnikové informační systémy: Podnik v informační společnosti. 2. vyd. Praha: Grada, 2008. 283 s. ISBN 978-80-247-2279-5.

HERNANDEZ, Michael J. Návrh databází. 1. vyd. Praha: Grada, 2006. 408 s. ISBN 80-247-0900-7.

MOLNÁR, Zdeněk. Efektivnost informačních systémů. 1. vyd. Praha: Grada, 2000. 144 s. ISBN 80-7169-410-X.

SODOMKA, Petr a Hana KLČOVÁ. Informační systémy v podnikové praxi. 2. vyd. Brno: Computer Press, 2010. 504 s. ISBN 978-80-251-2878-7.

Vedoucí bakalářské práce: Ing. Bernard Neuwirth, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2014/2015.

L.S.

doc. RNDr. Bedřich Půža, CSc.
Ředitel ústavu

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
Děkan fakulty

V Brně, dne 28.2.2015

Abstrakt

Hlavním předmětem bakalářské práce je optimalizace skladového informačního systému pro malé a střední firmy, který nabízí společnost M.C. software s.r.o. Vzhledem k rozsáhlosti této problematiky se tato práce bude zabývat pouze optimalizací databáze a uživatelského rozhraní. Výstup práce bude sloužit jako podklad pro vypracování nové verze tohoto informačního systému.

Abstract

The main subject of this bachelor thesis is optimalization of the stock management information system for small and middle enterprises which is offered by the company M.C. software s.r.o. Regarding the extent of this field this work will be dealing only with optimalization of the database and user interface. The output of the work will serve as the basis for elaboration of a new version of this information system.

Klíčová slova

informační systémy, optimalizace, databázové systémy, sklad, uživatelské rozhraní

Keywords

information system, optimalization, database system, stock, user interface

Bibliografická citace

ČMOK, P. *Návrh optimalizace dílčí části informačního systému*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2015. 69 s. Vedoucí bakalářské práce Ing. Bernard Neuwirth, Ph.D.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně.

Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb. O právu autorském a právech souvisejících s právem autorským).

V Brně 31. května 2015

.....

Petr Čmok

Poděkování

Tímto bych rád poděkoval vedoucímu mé bakalářské práce - panu Ing. Bernardu Neuwirthovi, Ph.D., za jeho odborné vedení, rady a cenné připomínky, které mi poskytl během vytváření práce.

Dále bych chtěl poděkovat zaměstnancům firmy M.C. software s.r.o., kteří mi poskytli všechny potřebné informace, vývojářské nástroje, zdrojové kódy a dokumentace potřebné k vytvoření této práce.

OBSAH

ÚVOD	10
CÍLE PRÁCE A METODIKA POSTUPU	11
1 TEORETICKÁ ČÁST PRÁCE	12
1.1 Informační systém	12
1.1.1 Data a informace	12
1.1.2 Definice systému	13
1.1.3 Definice informačního systému	14
1.1.4 Motivace vzniku informačního systému	14
1.1.5 Technologické pojetí informačního systému	15
1.1.6 Architektura podnikových informačních systémů	15
1.1.7 Klasifikace podnikových informačních systémů	16
1.1.8 Životní cyklus informačního systému	17
1.2 Aplikační vrstva	17
1.2.1 Ovládací prvky Windows Forms aplikace	18
1.3 Databázové systémy	21
1.3.1 Pojem databázový systém	21
1.3.2 Typy databázových systémů	22
1.3.3 Databázové modely	23
1.3.4 Relační databáze	24
2 ANALYTICKÁ ČÁST	29
2.1 Společnost M.C. software s.r.o.	29
2.1.1 Základní informace o společnosti	29
2.1.2 Sortiment služeb a zboží	29
2.1.3 Organizační struktura společnosti	31
2.1.4 Obchodní situace společnosti	31
2.1.5 Silné a slabé stránky společnosti	32

2.2	Skladový informační systém	32
2.2.1	Aplikační vrstva programu SKLAD	32
2.2.2	Databázová vrstva programu SKLAD	40
2.2.3	Okolí informačního systému	43
3	NÁVRH VLASTNÍHO ŘEŠENÍ	45
3.1	Vývojové prostředí a architektura	45
3.1.1	Vývojové prostředí a nástroje	45
3.1.2	Architektura informačního systému.....	45
3.2	Aplikační vrstva	45
3.2.1	Konzervativní přístup	45
3.2.2	Moderní přístup.....	49
3.3	Databázová vrstva	55
3.3.1	Návrh relačního modelu.....	56
3.3.2	Návrh SŘBD	58
3.4	Nástin modernějšího přístupu k problematice.....	58
3.5	Ekonomické zhodnocení práce	59
3.5.1	Přínosy navrhovaného řešení	59
3.5.2	Náklady řešení	60
	ZÁVĚR	62
	SEZNAM POUŽITÝCH ZDROJŮ	63
	SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ	66
	SEZNAM TABULEK, OBRÁZKŮ A GRAFŮ.....	67
	SEZNAM PŘÍLOH.....	69

ÚVOD

V této bakalářské práci budu vytvářet návrh optimalizace dílčí části skladového informačního systému, který nabízí společnost M.C. software s.r.o. Záměrně jsem si pro tuto práci vybral starší program z nabídky zmíněné společnosti, který dnes tvoří spíše doplňkovou službu pro zákazníky společnosti. Zajímavé bude porovnání programu, který byl v posledních patnácti letech pouze udržován bez výrazné aktualizace s moderními přístupy v této problematice, se kterými jsem se setkával při studiu.

Od práce očekávám, že mě obohatí o cenné zkušenosti z praxe se zaměřením na podnikové informační systémy a rozšíří mé dovednosti v oblastech softwarového návrhu a implementace.

CÍLE PRÁCE A METODIKA POSTUPU

Cílem bakalářské práce je provést návrh optimalizace nabízeného software ve společnosti M.C. software s.r.o.

Dílčím cílem práce bude provedení analýzy současného stavu informačního systému ve společnosti. Na základě výsledků analýzy bude následně vytvořen návrh optimálního řešení dílčí části informačního systému.

V první kapitole, teoretickém úvodu práce, nejprve ustanovím základní pojmy a moderní koncepce v problematice informačních systémů. V následující části práce se budu věnovat ovládacím prvkům, které budu využívat při tvorbě návrhu aplikační části programu. Zbývající část teoretické části budu věnovat databázovým systémům.

Ve druhé kapitole, kterou v mé práci bude představovat analytická část, nejprve představím zadavatelskou společnost – M.C. software s.r.o. Následně se budu věnovat analýze současného stavu skladového programu opět ze tří hledisek – aplikační část, databázová část a okolí informačního systému.

Třetí a zároveň nejdůležitější kapitola mé práce bude obsahovat návrh vlastního řešení, tedy návrh optimalizace skladového programu. Nejprve navrhnou společnosti vhodné koncepce a technologie pro aktualizaci tohoto programu, následně vytvořím návrh na přepracování aplikační části programu a optimalizace databázového systému. Práci zakončím ekonomickým zhodnocením celého projektu.

1 TEORETICKÁ ČÁST PRÁCE

Teoretickou část práce jsem rozdělil na 3 hlavní kapitoly – informační systémy, aplikační vrstvu a databázové systémy. Úvodní kapitola - informační systémy bude sloužit jako obecný úvod do této problematiky. Ve druhé části práce se budu zabývat vývojem WindowsForms aplikací v prostředí Microsoft Visual Studio 2013, ve kterém budu tvořit návrhy řešení této práce. V třetí kapitole teoretické části popíšu databázové systémy, jejich tvorbu a optimalizaci.

1.1 Informační systém

V této části práce se nejdříve zaměřím na základní pojmy, s jejichž pomocí ustanovím definici informačního systému. Následně odpovím na otázku, proč je pro každou společnost důležité se informačními systémy zabývat. Celou kapitolu zakončím základní klasifikací informačních systémů a popisem jejich životního cyklu.

1.1.1 Data a informace

Pojmy data a informace intuitivně používá každý celý život. Pro definování informačního systému je ale nutné, tyto pojmy přesně vymezit a pochopit rozdíl mezi nimi.

Data jsou ve světě informačních technologií chápána, jako čistá fakta. Tato data mohou mít různé formy, ať už se jedná o číslo zapsané na papír, nebo větu, převedenou na bity a bajty, které jsou uloženy v paměti počítače. Pro každou společnost je nutné zabezpečit proces pro zaznamenání těchto dat. [1]

S definicí dat úzce souvisí definice **informací**. Informace jsou data, která byla zpracována tak, že mají pro uživatele význam. Proces získávání informací tedy zahrnuje sběr dat, která jsou následně podrobena transformačnímu procesu za účelem vytvoření informací. Tento proces je možné vysvětlit i při pohledu na základní operace v běžném obchodě. Na pokladně se zaznamenávají data o prodeji jednotlivých položek. Tato data jsou následně zpracována a transformována na informace, které mohou poskytnout souhrnné informace o jednotlivých produktech – např. který produkt je nejprodávanější nebo které produkty se často nakupují společně. Na základě těchto informací je možné provést reorganizaci zboží, vybrat produkty do slevových akcí nebo zlepšit propagaci

obchodu. Z dat, která pro obchod neměla žádný význam, byly vytvořeny informace, které jsou užitečné při vedení podniku. [1]

Ačkoliv jsou informace bezesporu užitečnou surovinou pro jednotlivce i organizace, ne všechny informace jsou použitelné. Při rozlišení „dobrých“ a „špatných“ informací se musí zvážit vlastnosti spojené s časem, obsahem a formou, kterou byla informace podána. [1]

1.1.2 Definice systému

Slovo systém je často používané téměř v každém oboru a pravděpodobně proto neexistuje žádná přesná definice tohoto pojmu. V této práci systém definuji jako soubor částí, které spolupracují za účelem dosažení společného cíle. Ne každý systém má pouze jeden cíl a často systém obsahuje několik podsystémů, které mají své vlastní dílčí cíle. [2]

Příkladem systému může být počítač. Na nejnižší úrovni se skládá z tranzistorů, rezistorů a dalších miniaturních součástí, které se slučují do podsystémů - grafická karta, základní deska atd. Tyto podsystémy jsou závislé na dalších podsystémech – na podsystému chlazení a podsystému rozvodů napájení ze zdroje. Dohromady tyto podsystémy tvoří systém – počítač. Pokud by byl rozebrán na jednotlivé podsystémy, přestane fungovat.

1.1.2.1 Klasifikace systémů

Podle interakce s okolím se systémy rozlišují na otevřené a uzavřené. Otevřené systémy se vyznačují nutnou interakcí s okolím systému. Uzavřené systémy naopak nejsou ovlivňovány svým okolím. [2]

Podle chování systému jsou klasifikovány na deterministické systémy a stochastické systémy. Deterministické systémy mají jednoznačně dané chování – pokud by byl do deterministického systému opakovaně vložen stejný vstup, výsledek by byl vždy stejný. Toto nelze říci o stochastických systémech, ve kterých mají vliv náhodné veličiny. [2]

Podle vývoje v čase se systémy dělí na statické a dynamické. Pojmem statické systémy lze označit ty systémy, které se v čase nevyvíjí. Oproti tomu se dynamické systémy

v čase mění, ať už jde o úpravy za účelem vylepšení výstupů ze systému nebo nutné reakce na změny v okolí systému. [2]

1.1.3 Definice informačního systému

Na termín informační systém existuje mnoho různých pohledů. Jedna z možných definic informačního systému ho popisuje jako soubor lidí, technologických prostředků a metod, které spolu zajišťují sběr, přenos, zpracování a uchování dat, za účelem prezentování informací pro potřeby uživatelů. [3]

Takto definovaný informační systém nemusí být nutně automatizovaný pomocí počítačů, ale může se vyskytovat i v např. papírové podobě. Nutno však podotknout, že trendem posledních dvaceti let je digitalizace dat a automatizace informačních systémů za pomoci počítačů a moderních technologií. Příkladem informačního systému může být telefonní seznam, kartotéka nebo účetnictví.

S pojmem informační systém úzce souvisí pojem informační a komunikační technologie (dále jen ICT, z anglického information and communications technology). Pod tento pojem se řadí všechny komunikační zařízení a aplikace. Jako příklad lze uvést rádio, televizi, telefony, počítače, satelitní systémy i aplikace spojené s těmito technologiemi jako videokonference, nebo odborné fórum. [4]

1.1.4 Motivace vzniku informačního systému

Informační systémy a informační technologie hrají podstatnou roli v každodenním provozu hospodářských subjektů a mohou výrazně zlepšit jejich konkurenceschopnost, neboť kdykoliv umožňují:

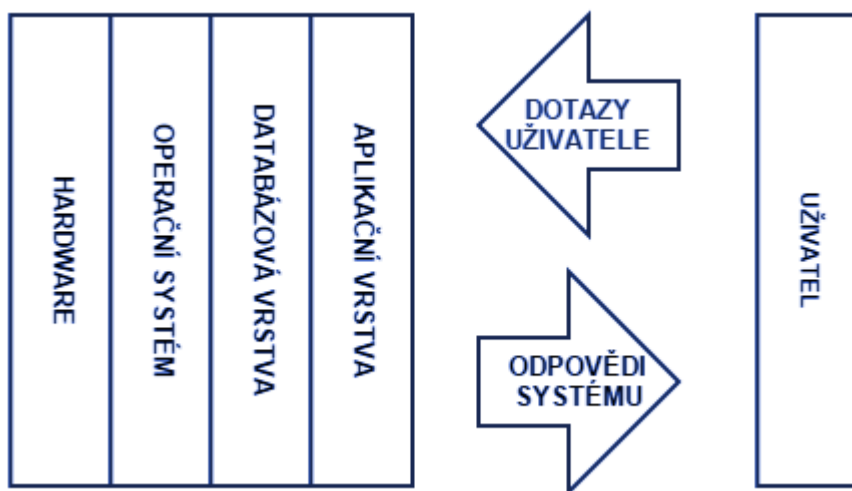
- rychlé získávání, zpracování a předání informace,
- nové a efektivnější způsoby komunikace se zákazníky a partnery,
- podporu personalizace produkce v hromadné výrobě,
- samoobslužné procesy,
- kvalitnější způsoby uchování dat pomocí jejich digitalizace,
- prodej po internetu. [5]

Informatizace společnosti zvyšuje kvalifikaci obyvatel v IS/ICT a tak zvyšuje výkonnost ekonomiky a generuje nové trhy. Dobrý byznys manažer musí umět využívat ICT nejen pro svou potřebu, ale zejména pro efektní řízení firmy a jejich procesů.

S ohledem na roli ICT v současné ekonomice porostou i v tomto desetiletí investice podniků do ICT. Stejně tak poroste poptávka po specialitech, kteří budou disponovat ICT znalostmi a současně znalostmi ekonomických procesů. [5]

1.1.5 Technologické pojetí informačního systému

Tradiční informační systém má z technologického hlediska pět vrstev – hardware, operační systém, databázovou platformu, aplikační software a uživatele. Uživatel komunikuje s celým systémem prostřednictvím aplikační vrstvy. Komunikace mezi dalšími vrstvami je plně automatická. [6]



Obrázek 1 - Technologické pojetí informačního systému [6]

1.1.6 Architektura podnikových informačních systémů

Dvouvrstvá architektura s výkonem soustředěným u klienta – veškeré aplikační a uživatelské služby se zpracovávají na klientské stanici uživatele. Toto uspořádání bývá také označováno jako tlustý klient. Při aplikaci této architektury je nutné brát v úvahu, že mezi klientem a serverem musí probíhat velký počet datových přenosů. [6]

Dvouvrstvá architektura s výkonem soustředěným na serveru – na klientskou stanici se přesouvají pouze uživatelské služby a vyžádané údaje. Toto uspořádání bývá také označováno jako tenký klient. Aplikační a datové služby zpracovává server. Je tedy nutné počítat s vysokými požadavky na jeho výpočetní sílu. [6]

Třívrstvá architektura – klient pracuje pouze s uživatelským rozhraním. Datové a aplikační služby jsou od sebe odděleny do samostatných logických celků. Třívrstvý

model vykazuje vyšší stabilitu, neboť je zátěž provozu celého řešení rozložena na více než jeden server. [6]

N-vrstvá architektura – podobně jako u třívrstvé architektury – informační systém je rozdělen do menších logických celků, které jsou rozmístěny na více serverových zařízeních. [6]

Moderním trendem v rozvoji architektur je **virtualizace**, která umožňuje k dostupným zdrojům přistupovat jiným způsobem, než jsou fyzicky uspořádány. Je tak mnohem jednodušší přizpůsobovat prostředí uživatelským potřebám. Mezi hlavní přínosy virtualizace patří snižování nákladů při pořizování hardwaru, úspory energie, zvýšení efektivity správy IS/ICT a otevření nových možností a postupů. [6]

1.1.7 Klasifikace podnikových informačních systémů

Trendem v informačních systémech je přechod od funkcionálního rozdělení informačních systémů na holisticko-procesní rozdělení. [6]

1.1.7.1 Funkcionální přístup

Jedná se o tradiční rozdělení aplikací v informačním systému společnosti, kde každý software (nebo modul softwaru) zastává určitou funkci. Informační systém společnosti se může například skládat z účetního IS, IS na evidenci zásob, mzdového IS a dalších IS, které zastávají jasně definovanou funkci. [6]

1.1.7.2 Holisticko-procesní přístup

„Podnikové informační systémy je vhodné klasifikovat podle jejich praktického uplatnění, ve shodě s nabídkou dodavatelů a shodě s požadavky na řízení podnikových procesů. Rozhodují pro klasifikaci podnikových informačních systémů je tzv. holisticko-procesní pohled.“ [6]

- ERP (Enterprise Resource Planning) je jádro informačního systému zaměřené na řízení interních podnikových procesů.
- CRM (Customer Relationship Management) systém obsluhuje procesy, které jsou směřované k zákazníkům.

- SCM (Supply Chain Management) systém řídí dodavatelský řetězec. Součástí tohoto systému bývá i systém sloužící k pokročilému plánování a rozvrhování výroby.
- MIS (Management Information Systems) systém pro podporu manažerského rozhodování. Sbírá data z ERP, CRM a SCM systémů a na jejich základě poskytuje informace pro podporu rozhodovacího procesu podnikovému managementu. [6], [7]

1.1.8 Životní cyklus informačního systému

Životní cyklus informačního systému lze chápat, jako za sebou jdoucí období s jasně definovaným cílem. K dosažení tohoto cíle jsou směřovány veškeré činnosti související s daným informačním systémem. [8]

Hlavní etapy životního cyklu informačního systému se různí dle autorů a neexistuje na ně jednoznačný pohled. Takto jednotlivé etapy rozděluje RNDr. JUDr. Vladimír Šmíd, CSc.:

- předběžná analýza neboli specifikace cílů,
- analýza systému neboli specifikace požadavků,
- projektová studie neboli vytvoření návrhu,
- implementace,
- testování,
- zavádění systému,
- zkušební provoz,
- rutinní provoz a údržba,
- reengineering. [8]

V praxi se tyto etapy velice často přizpůsobují společnosti, pro kterou je informační systém implementován. [8]

1.2 Aplikační vrstva

Vzhledem k tomu, že součástí této práce není implementace projektu, budu se v této kapitole zabývat pouze základními ovládacími prvky, které jsou součástí vývojového

prostředí Microsoft Visual Studio 2013, ve kterém budu vytvářet návrh nového uživatelského rozhraní.

1.2.1 Ovládací prvky Windows Forms aplikace

Funkcionální výčet nástrojů, které má vývojář k dispozici po nainstalování prostředí Microsoft Visual Studio 2013:

- zobrazí dat:
 - DataGridView – přizpůsobitelné tabulky, základní funkce pro jejich editaci;
- vázání dat a navigace:
 - BindingSource – zjednodušuje propojení ovládacích prvků formuláře a dat;
 - BindingNavigator – poskytuje rozhraní pro manipulaci s daty ve formuláři;
- úpravy textů:
 - TextBox – zobrazí upravitelný text v programu;
 - RichTextBox – zobrazí upravitelný text v programu, umožňuje základní formátování;
 - MaskedTextBox – TextBox s omezeným formátem pro zadaný text;
- zobrazení informací:
 - Label – zobrazí text, který uživatel nemůže změnit;
 - LinkLabel – zobrazí text, který uživatel nemůže změnit, po kliknutí otevře jiné okno programu;
 - StatusStrip – zobrazí informace o aktuálním stavu aplikace (obvykle jako stavový řádek v dolní části aplikace);
 - ProgressBar – zobrazí aktuální průběh operace;
- zobrazení webové stránky:
 - WebBrowser – umožňuje procházení webových stránek uvnitř formuláře;
- výběr ze seznamu:

- `CheckedListBox` – posuvný seznam se zaškrťovacími políčky;
- `ComboBox` – rozevírací seznam položek;
- `DomainUpDown` – seznam textových položek, které uživatel prochází pomocí tlačítek nahoru a dolů;
- `ListBox` – seznam textu a grafických položek;
- `ListView` – zobrazí položky v jednom ze čtyř možných zobrazení (pouze text, text a malé ikony, text a velké ikony či podrobné zobrazení);
- `NumericUpDown` – seznam čísl, které uživatel vybere pomocí tlačítka nahoru a dolů;
- `TreeView` – hierarchická kolekce objektů;
- obrázky a grafika:
 - `PictureBox` – zobrazení grafických souborů v rámečku;
 - `ImageList` – slouží jako úložiště pro obrázky;
- nastavení hodnoty:
 - `CheckBox` – zaškrťovací políčko s popisujícím textem;
 - `CheckedBoxList` – posuvný seznam se zaškrťovacími políčky;
 - `RadioButton` – `CheckedBoxList` s možností výběru pouze jedné položky;
 - `TrackBar` – umožňuje nastavit hodnoty na posuvné stupnici;
- nastavení data:
 - `DateTimePicker` – graficky zobrazí kalendář, na kterém je možné vybrat datum a čas;
 - `MonthCalendar` – zobrazí grafický kalendář, který umožní vybrat rozsah dat;
- dialogová okna:
 - `ColorDialog` – slouží k výběru barvy;

- FontDialog – slouží k výběru fontu a jeho atributů;
- OpenFileDialog – umožní uživateli vyhledat a vybrat soubor;
- PrintDialog – umožní uživateli vybrat tiskárnu a nastavit její atributy;
- PrintPreviewDialog – zobrazí náhled tisku;
- FolderBrowserDialog – umožní uživateli najít a vybrat složku;
- SaveFileDialog - umožní uživateli uložení souboru;
- ovládací prvky nabídky:
 - MenuStrip – umožní vytvoření nabídky (menu);
 - ContextMenuStrip – umožní vytvoření lokální nabídky, která se zobrazí po stisknutí pravého tlačítka myši;
- příkazy:
 - Button – klasické tlačítko pro spuštění nebo přerušení procesu;
 - NotifyIcon – využití notifikační oblasti operačního systému;
 - ToolStrip – vytvoří panel nástrojů;
- nápověda pro uživatele:
 - HelpProvider – poskytuje „vyskakovací“ nápovědu;
 - ToolTip – poskytuje informace o účelu ovládací prvku, zobrazí se po zanechání kurzoru myši na ovládacím prvku;
- seskupování jiných ovládacích prvků:
 - Panel – vytvoří skupinu ovládacích prvků bez popisu, posuvník pokud je nutný;
 - GroupBox – vytvoří skupinu ovládacích prvků s popisem, bez posuvníku;
 - TabControl – vytvoří skupiny ovládacích prvků, společně s nabídkou, kde uživatel může vybrat, kterou skupinu chce právě zobrazit;

- SplitContainer – rozdělí část formuláře na skupinu panelů;
- TableLayoutPanel – skupina panelů v mřížce, která se dynamicky upravuje;
- FlowLayoutPanel – panel, který se dynamicky upravuje vodorovně nebo svisle;
- zvuk:
 - SoundPlayer – přehrává zvukové efekty ve formátu WAV. [9]

1.3 Databázové systémy

Optimalizace databázového systému bude tvořit významnou část práce. V této části se budu zabývat základními pojmy spojenými s tímto tématem, obecnou klasifikací databází a databázovými modely. Následně se blíže zaměřím na problematiku relačních databází.

1.3.1 Pojem databázový systém

Databázové systémy se skládají z databáze a systému řízení báze dat. [10]

1.3.1.1 Databáze

Pojem databáze v informačních technologiích lze chápat jako uspořádanou sdílenou množinu **dat** a **metadat**, která je uložena na paměťovém médiu. Metadata popisují základní charakteristiky a principy uvnitř databáze. Společně se softwarovými prostředky SŘBD tvoří databázový systém. [10]

1.3.1.2 Systém řízení báze dat

Systém řízení báze dat (dále jen SŘBD) je soubor programů, které spravují databázovou strukturu a dovolují získávat potřebné informace v přehledné a dobře organizované formě. Používání SŘBD oproti přímému napojení databáze na aplikace přináší mnoho výhod:

- **Lepší sdílení dat** – SŘBD pomáhá vytvořit funkční prostředí, ke kterému může více aplikací přistupovat k jedné databázi.
- **Lepší přizpůsobivost koncových aplikací** – kvalitně navrhnutá vrstva SŘBD usnadní programátorům vytváření a přizpůsobování aplikací pro koncového uživatele.

- **Lepší integrita dat** – SŘBD umožňuje kontrolovat integritu dat centralizovaně na jednom místě a tak zabraňuje nedostatečným kontrolám, které by se mohly objevovat u jednotlivých aplikací přistupujících k databázi.
- **Větší produktivita systému** - rozdělení zátěže na hardware mezi uživatelskou stanicí a server [10]

1.3.2 Typy databázových systémů

Databázové systémy je možné rozdělit podle mnoha faktorů:

1.3.2.1 Dle počtu uživatelů

Single-user databáze podporují pouze jednoho uživatele v daný okamžik – pokud uživatel A operuje s databází, uživatel B a C musí čekat, než uživatel A dokončí svou činnost. [10]

Multiuser databáze podporuje více uživatelů v daný okamžik. Pokud se jedná o skupinu zaměstnanců do 50 uživatelů, databáze zároveň spadá pod označení **workgroup databáze**. Pokud je databáze používána více než 50 uživateli současně, jedná se o **enterprise databázi**. [10]

1.3.2.2 Dle lokace

Jako **centralizovanou databázi** lze označit takovou, ve které jsou všechna data uložena na jednom místě. Mezi hlavní výhody centralizované databáze bezesporu patří snadnější přístup vedení k datům a analytickým službám, které následně zlepšují rozhodovací proces ve společnosti. [10]

Distribuované databáze jsou naproti tomu umístěné na několika místech, většinou tam, kde jsou nejčastěji využívány. Výhodou tohoto řešení je nízká náročnost na hardware na serverové jednotce. [10]

1.3.2.3 Dle využití

Operační databázi lze označit takovou databází, která je zaměřena a optimalizována na běžné provozní operace - vkládání nových dat, úpravu a mazání existujících dat. [10]

Datové sklady jsou oproti operačním databázím velice rozsáhle databáze, využívané pro získání souhrnných analytických informací v reálném čase. [10]

1.3.3 Databázové modely

Z hlediska způsobu uložení dat a vazeb, které mezi nimi vznikají, je možné databáze rozdělit do základních modelů:

1.3.3.1 Hierarchický model

Jak již název modelu napovídá, data jsou zde strukturována hierarchicky a obvykle se znázorňují jako obrácený strom. Základní tabulka modelu slouží jako „kořen“ obráceného stromu a ostatní tabulky z něj vycházejí jako větve. Vztahy v hierarchickém modelu jsou reprezentovány termíny rodič a potomek. Uživatel k datům v hierarchickém modelu může přistupovat tak, že začne v kořenové tabulce a postupně se přes stromovou strukturu propracovává k hledaným datům. Nevýhodou tohoto systému je, že uživatel musí dobře znát celou strukturu databáze. [11]

1.3.3.2 Síťový model

Síťový model je v podstatě zobecnění a vylepšení hierarchického modelu. Struktura v síťové databázi je vyjádřena pomocí uzlů a množinových struktur. Uzel zde reprezentuje soubor záznamů a množinová struktura reprezentuje vztah v síťové databázi. Jeden uzel je zde definován jako vlastník a druhý jako prvek. Hlavní výhodou oproti hierarchickému modelu je fakt, že jeden vlastník může mít více prvků. Mezi dvěma uzly může být definováno jedno nebo více spojení a libovolný uzel může být součástí dalších množin s jinými uzly v databázi. [11]

1.3.3.3 Relační model

Základem je matematické zobecnění pojmu soubor pomocí silného formalizmu matematické relace a využívání operací relační algebry. Model má jednoduchou strukturu, data jsou organizována v tabulkách (relacích), které se skládají z řádků (záznamů) a sloupců (atributů). Podstatnou roli v tomto modelu hrají vztahy, mezi jednotlivými relacemi. [11]

1.3.3.4 Objektový model

Objektově orientovaný datový model obsahuje všechny charakteristiky objektově orientovaných jazyků. Relační databáze se při použití tohoto modelu degraduje na pouhé skladiště dat. Hlavní myšlenkou objektově orientovaného modelu je, že databázový vývojář se stará o všechny aspekty databáze z objektově orientovaného

programu. Neexistuje tak tedy jasné rozdělení mezi databázovým programem a aplikačním softwarem. [11]

1.3.3.5 Objektově relační model

Jedná se o rozšíření relačního modelu o softwarovou vrstvu, dodávající relačnímu SŘBD objektový charakter a bohatší typový systém. Mezi hlavní výhody tohoto modelu patří možnost využití tříd, zapouzdření a dědičnosti. Cílem pro vytvoření tohoto modelu bylo, aby relační databáze dokázala zpracovávat i složitější datové typy, jako například audio záznamy, video nebo složité stavební nákresy. Tento model také bývá označován jako rozšířený relační datový model. [11]

1.3.4 Relační databáze

Relační databáze byla poprvé představena v roce 1969 pracovníkem firmy IBM - Dr. E. F. Codd a v současnosti je bezesporu nejrozšířenější model používaný při správě databází. [11]

1.3.4.1 Struktura databáze

V problematice relačních databází je velice důležité správné názvosloví. Relační databázový model sdružuje data do **relací** (tabulek). Každá tato relace se skládá z **atributů** (sloupců tabulky) a **záznamů** neboli **n-tic** (řádky tabulky). Každý atribut relace má definovaný jednoznačný název, typ a rozsah. **Hodnotou** nazýváme prvek daného atributu a záznamu. Se strukturou databáze úzce souvisí **pohledy**. Pohledy jsou virtuální databázové tabulky, které vznikly spojením polí z jedné nebo více tabulek. [11], [12]

1.3.4.2 Klíče

Klíče jsou ve struktuře tabulky zásadním prvkem:

- zabezpečují, že každý záznam v tabulce je přesně identifikován,
- napomáhají dodržení různých druhů integrity,
- slouží jako prostředek pro vytváření vztahů. [11]

Rozlišují se následující druhy klíčů:

Kandidát na klíč je jedno nebo více polí, které jednoznačně identifikují jednotlivé instance entity tabulky. Ze všech možných kandidátů na klíč vybereme jednoho a označíme ho jako primární klíč. [11]

Vlastnosti kandidáta na klíč:

- Nesmí být vícesložkové pole.
- Musí obsahovat jedinečné hodnoty.
- Nesmí obsahovat hodnoty null.
- Jeho hodnota nesmí být tajná (její únik nesmí představovat bezpečnostní hrozbu).
- Hodnota nesmí být volitelná.
- Skládá se z nejmenšího možného počtu polí, které zaručí jedinečnost.
- Jeho hodnota jednoznačně identifikuje hodnoty všech polí daného záznamu.
- Jeho hodnota se mění pouze ve výjimečných případech. [11]

Kandidátní klíč může být uměle vytvořen. Kandidáti na klíč, kteří se nestanou primárním klíčem, se označují jako **alternativní klíče**. [11]

Primární klíč má za úkol jednoznačně identifikovat tabulky v rámci celé databázové struktury a pomáhá při vytváření vazeb mezi tabulkami. Hodnota primárního klíče jednoznačně identifikuje daný záznam v rámci tabulky a také v rámci celé databáze. Každá tabulka musí mít pouze jeden primární klíč, který je vybrán z kandidátních klíčů. [11]

Při výběru primárního klíče je vhodné vybrat kandidáta, který je definován co nejmenším počtem polí a zároveň ve svém jméně obsahuje i jméno tabulky. Primární klíč složený z více než jednoho pole se nazývá **složený primární klíč**. [11]

Atribut relace se může nazvat **cizím klíčem**, pokud splňuje tyto vlastnosti:

- Každá hodnota cizího klíče je buď plně zadaná, nebo plně nezadaná.

- Existuje relace R1 s kandidátním klíčem takovým, že hodnota cizího klíče je identická s hodnotou kandidátního klíče relace R1.

Soulad hodnot cizích a primárních klíčů představuje vztahy mezi řádky tabulek - „drží“ databázi pohromadě. [11]

Ostatní atributy relace se nazývají **neklíč**. [11]

1.3.4.3 Vztahy mezi relacemi

Vztahy mezi jednotlivými relacemi nejsou vždy stejné. Základní vlastnosti, které popisují tyto vztahy, jsou: stupeň relace, kardinalita vztahu a volitelnost účasti ve vztahu. Ke znázornění těchto vlastností slouží Entity-Relationship diagram (ER diagram).

Stupeň vztahu definuje počet účastníků ve vztahu. Rozlišujeme následující typy vztahů:

- unární - vztah sama na sebe,
- binární - vztah mezi dvěma relacemi,
- ternární - vztah mezi třemi relacemi,
- n-ární - vztah mezi n-relacemi zároveň. [13]

V praxi se s ternárními a n-árními stupni vztahu setkáváme velice zřídka. [13]

Druhou vlastností, která určuje podobu vztahu mezi relacemi, je **kardinalita vztahu**. Ta může nabývat následujících hodnot:

- Mezi tabulkami není žádná spojitost.
- Vztah 1:1 - používáme, pokud záznamu odpovídá právě jeden záznam v jiné databázové tabulce a naopak. Takovýto vztah je používán pouze ojediněle, protože většinou není pádný důvod, proč takovéto záznamy neumístit do jedné databázové tabulky.
- Vztah 1:N - přiřazuje jednomu záznamu více záznamů z jiné tabulky. Jedná se o nejpoužívanější typ relace, jelikož odpovídá mnoha situacím v reálném životě.

- Vztah M:N - umožňuje několika záznamům z jedné tabulky přiřadit několik záznamů z tabulky druhé. V databázové praxi bývá tento vztah z praktických důvodů nejčastěji realizován kombinací dvou vztahů 1:N a 1:M. [13]

Poslední vlastností, která definuje vztah mezi relacemi, je **volitelnost účasti ve vztahu**, která určuje, zda je účast ve vztahu povinná, nebo volitelná. [13]

1.3.4.4 Integrita databáze

V problematice relačních databází se rozlišují následující druhy integritních omezení:

- **Doménové integritní omezení** definuje platné hodnoty daného atributu. Pod toto omezení spadá výběr logického datového typu, definování měřítka, definování přesnosti a nakládání s neznámými a neexistujícími hodnotami.
- Omezení **přechodové integrity** definují stavy, kterým mohou dané hodnoty předcházet (např. rodinný stav svobodný, se nemůže změnit na rozvedený, ale osobě, která má rodinný stav rozvedený, zpravidla musí předcházet rodinný stav ženatý/vdaná).
- **Entitní integritní omezení** zajišťuje, aby nebylo možné vložit do databáze duplicitní záznamy. Jedná se tedy o zajištění unikátnosti skutečných identifikátorů reálných objektů. Za entitní omezení lze považovat existenci správně navrženého primárního klíče.
- **Referenční integrita** má za úkol ochraňovat vazby mezi relacemi. V praxi to znamená, že žádný cizí klíč, se nesmí stát „sirotkem“ – žádný řádek v cizí tabulce nesmí obsahovat takovou hodnotu cizího klíče, která nemá odpovídající záznam v primární tabulce.
- **Databázová integrita** zachycuje omezení, které se týkají obsahu databáze (např. zákazník nemůže být členem zvýhodněné skupiny, pokud v posledních 6 měsících nevytvořil objednávku).
- Omezení **transakční integrity** ovládají způsoby přípustné manipulace s daty (např. transakce v bankovníctví – částka se odečte z účtu A, ale nepodaří se ji připsat na účet B. V takovém případě se peníze vrátí na účet A.). Jsou procedurálního charakteru a nejsou součástí datového modelu. [13]

Vývojář má dnes velké množství nástrojů, jejichž pomocí může integritu databáze zajistit. Mezi ty nejvýznamnější a nejpoužívanější patří tzv. **triggery**, neboli česky spouště. Ty definují činnosti, které se mají provést v případě definované události nad databázovou tabulkou. [14]

1.3.4.5 Normální formy

První normální forma - relace je v první normální formě, právě když všechny její jednoduché domény obsahují pouze atomické hodnoty. [12]

Druhá normální forma - relace je ve druhé normální formě, právě když je v 1NF a každý její neklíčový atribut, je plně funkčně závislý na každém kandidátním klíči. [12]

Třetí normální forma - relace je ve třetí normální formě, právě když je ve 2NF a neexistuje žádný neklíčový atribut, který je tranzitivně závislý na některém kandidátním klíči. [12]

Existuje také čtvrté normální forma, označovaná také jako Boyce Codd normální forma, a pátá normální forma, ty jsou ovšem při praktickém návrhu zvažovány jen zřídka. Nepřihlížení k těmto pravidlům může vést k poněkud méně dokonalému návrhu databáze, ale nemají žádný vliv na funkčnost databáze. [15]

2 ANALYTICKÁ ČÁST

V této části práce nejprve představím společnost M.C. software s.r.o., pro kterou je tato práce tvořena. Následně se budu podrobně zabývat současným stavem informačního systému, který se bude vylepšovat.

2.1 Společnost M.C. software s.r.o.

2.1.1 Základní informace o společnosti

Název: M.C. software s.r.o.

Sídlo: Humpolec, Kamarytova 161, okres Pelhřimov, PSČ 39601

Právní forma: Společnost s ručením omezeným, zapsáno 28. ledna 1994

Předmět podnikání:

- koupě zboží za účelem jeho dalšího prodeje a prodej,
- poskytování software a poradenství v oblasti hardware a software,
- výuka jazyků,
- činnost podnikatelských, finančních, organizačních a ekonomických poradců,
- činnost účetních poradců, vedení účetnictví, vedení daňové evidence,
- překladatelská a tlumočnická činnost,
- výroba, instalace a opravy elektronických zařízení.

Jednatel: ing. MIROSLAV ČMOK [16]

2.1.2 Sortiment služeb a zboží:

2.1.2.1 Poskytování software

Společnost M.C. software s.r.o. nabízí funkcionálně orientované informační systémy, které řeší následující oblasti:

Účetnictví

„Program MC-EKON je určen pro menší a střední firmy. Umožňuje účtování na střediska, zakázky. Lze vytisknout mimo jiné výsledovku, rozvahu, Cash Flow.“ [17]

Dlouhodobý a drobný majetek

„Slouží ke sledování dlouhodobého a drobného majetku. Počítá odpisy, tiskne inventury a karty majetku“ [17]

Mzdy zaměstnanců

„Mzdový program je určen pro menší společnosti s několika zaměstnanci, ale dobře poslouží i větším firmám. Samozřejmostí jsou výpočty náhrad za pracovní neschopnost, ELDP, mzdové listy, atd. Program překvapí jednoduchým ovládáním a přehledností.“ [17]

Odbyt

„Program odbyt je určen pro velkoobchod. Umožňuje sledování objednávek zákazníků, generování dodacích listů a faktur...“ [17]

Program pro pekárny a cukrárny

„Rozsáhlý projekt, který pokrývá celou oblast od evidence surovin, přes výrobu až po fakturaci. Bohaté statistiky. Díky tomuto programu se můžeme považovat za specialisty na pekárny a cukrárny.“ [17]

Maloobchod

„Pro maloobchod je určen program MARKET. Řeší skladové hospodářství prodejny, správu ceníků, akční ceny, komunikaci s registrační pokladnou. Umožňuje prodej i přímo z počítače.“ [17]

Spedice

„Evidence objednávek spediční firmy, evidence zákazníků, dopravců... Od objednávky až po fakturu.“ [17]

Evidence docházky

„Jeden z našich nejmladších programů. Jedná se o komplexní řešení sledování docházky. V nabídce je hardware - přístupové terminály a součástí řešení je program DOCHÁZKA.“ [17]

Sklad

„Program umožňuje evidenci skladových zásob, a to i na více skladech. Dále slouží k vystavování příjmových a výdajových dokladů, udržování ceníků materiálu, atd.“ [17]

Společnost nabízí a vývoj **software na zakázku**. [17]

2.1.2.2 Poskytování hardware

- Prodej hardware,

- poskytování základní podpory,
- hardwarové poradenství,
- správa sítě,
- registrační poklady. [17]

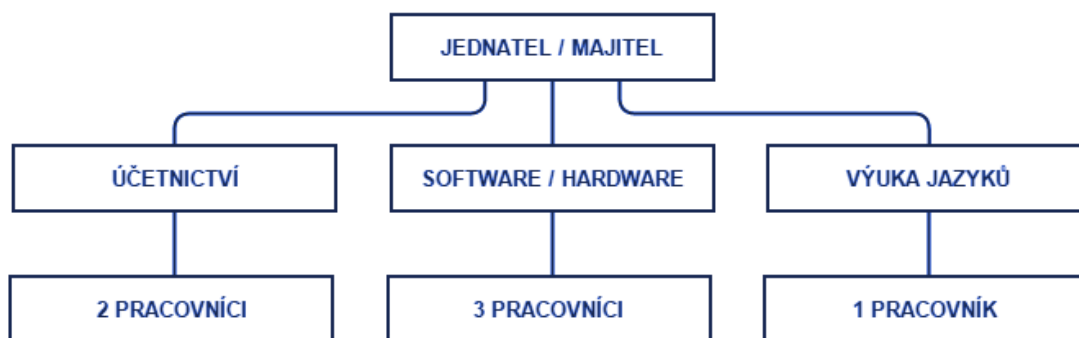
2.1.2.3 Vedení účetnictví

- Zpracování účetnictví,
- zpracování mezd,
- daňové poradenství,
- účetní poradenství. [17]

2.1.2.4 Výuka jazyků

- Angličtina. [17]

2.1.3 Organizační struktura společnosti



Obrázek 2 - Organizační struktura společnosti M.C. software s.r.o.

2.1.4 Obchodní situace společnosti

Největší podíl zákazníků společnosti tvoří právnické osoby – zejména pak malé a střední firmy, které si buď nechávají zpracovávat kompletní účetnictví přímo ve společnosti M.C. software s.r.o., nebo si koupí licence účetních a jiných programů a následně využívají podporu společnosti M.C. software s.r.o. Prodej hardware, poskytování software a vedení účetnictví a mezd tvoří zhruba stejný podíl na tržbách, výuka angličtiny má podíl na celkových tržbách 14%. Software a účetní služby společnost poskytuje po celém území České republiky, naproti tomu výuka angličtiny a prodej hardware má zákazníky převážně v kraji Vysočina. [18]

2.1.5 Silné a slabé stránky společnosti

2.1.5.1 Silné stránky společnosti:

- Dobré jméno společnosti,
- spokojení dlouholetí klienti,
- dobrá propagace na internetu,
- strategické umístění podniku uprostřed ČR,
- působení společnosti ve více odvětvích,
- produkty jsou vyladěny dlouhodobou zpětnou vazbou uživatelů,
- poskytuje kompletní služby,
- podpora programů přímo od vývojářů.

2.1.5.2 Slabé stránky společnosti:

- Zastaralé technologie při vývoji některých programů,
- náhodné získávání nových klientů.

2.1.5.3 Příležitosti:

- Rozšíření výuky jazyků,
- větší propagace sítí s certifikátem LYNX Cabling Systém,
- přijetí zaměstnance zabývajícího se prodejem a propagací.

2.1.5.4 Hrozby

- Nová konkurence,
- zastaralé technologie mohou způsobit problémy s novými operačními systémy,
- umístění firemního serveru.

2.2 Skladový informační systém

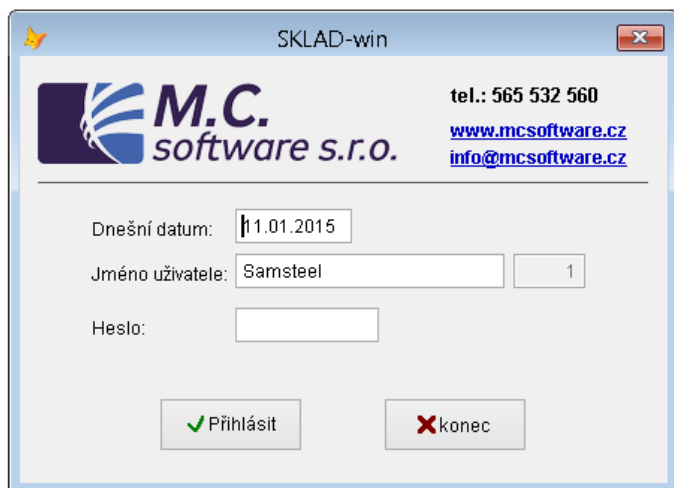
V této bakalářské práci budu hledat způsoby, jakými by bylo možné vylepšit skladový informační systém, který společnost M.C. software s.r.o. vyvíjí. Nejdříve popíšu, co současná verze skladového programu poskytuje, následně analyzuji současné uživatelské rozhraní, databázové systémy a okolí tohoto informačního systému, kde se budu zabývat převážně legislativou spojenou s touto tematikou a zhodnocením technologií použitých při vývoji programu.

2.2.1 Aplikační vrstva programu SKLAD

V této části práce jsou popsány základní formuláře skladového programu.

2.2.1.1 Autorizace uživatele

Po spuštění skladového programu se spustí první formulář, který po uživateli vyžaduje přihlašovací údaje a možnost nastavit datum, se kterým bude program pracovat jako s výchozím nastavením pro případ, že potřebujeme vytvořit doklad s jiným než dnešním datem.

The image shows a Windows-style application window titled "SKLAD-win". Inside the window, at the top left, is the logo for "M.C. software s.r.o." with a stylized blue and white graphic. To the right of the logo, the text "tel.: 565 532 560" is displayed, followed by two lines of blue hyperlinks: "www.mcsoftware.cz" and "info@mcsoftware.cz". Below this header section, there are three input fields. The first is labeled "Dnešní datum:" and contains the text "1.01.2015". The second is labeled "Jméno uživatele:" and contains the text "Samsteel". To the right of this field is a small box containing the number "1". The third field is labeled "Heslo:" and is currently empty. At the bottom of the window, there are two buttons: a green button with a checkmark icon and the text "Přihlásit", and a red button with an "X" icon and the text "konec".

Obrázek 3 - Přihlašovací formulář do původní verze programu

Na tomto formuláři není prakticky co vylepšovat – můžeme zde hovořit pouze o detailech – např. proč se zobrazuje ID uživatele (TextBox vpravo od jména uživatele) nebo proč TextBox hesla a uživatelského jména není stejně dlouhý, jak to bývá běžné u jiných autorizačních formulářů. Společnost by také měla sjednotit psaní velkých nebo malých písmen na tlačítkách formuláře (na tlačítkách přihlásit a konec tomu tak není). Také by stála za úvahu možnost stálého přihlášení. Na druhou stranu možnost změny výchozího data hned při startu programu je dle mého názoru dobrým řešením.

2.2.1.2 Základní nabídka programu

Po úspěšném přihlášení uživatele se zobrazí formulář s nabídkou základních akcí. Zde může uživatel zvolit spuštění hlavního rozhraní skladového programu, nebo některou z netypických akcí:

- Uzavřít období / vrátit uzávěrku – program pracuje s účetními obdobími. Na konci každého účetního období se provede akce „uzavřít období“. Program následně provede kontrolní součty, pokud je vše v pořádku otevře nové účetní období.

- Spustit zálohu dat / obnovit data ze zálohy – vytvoří kompletní zálohu celé databáze do souboru. V případě potřeby se tato záloha může obnovit.
- Provést reindexaci – servisní akce, která ověří integritu databáze.
- Opravit struktury – funkce synchronizuje verzi databázového systému s verzí aplikace. Jedná se o servisní akci.
- Nahrát data z DOS skladu – export dat ze starší verze programu.
- Spravovat uživatelská práva.



Obrázek 4 - Základní nabídka původní verze programu

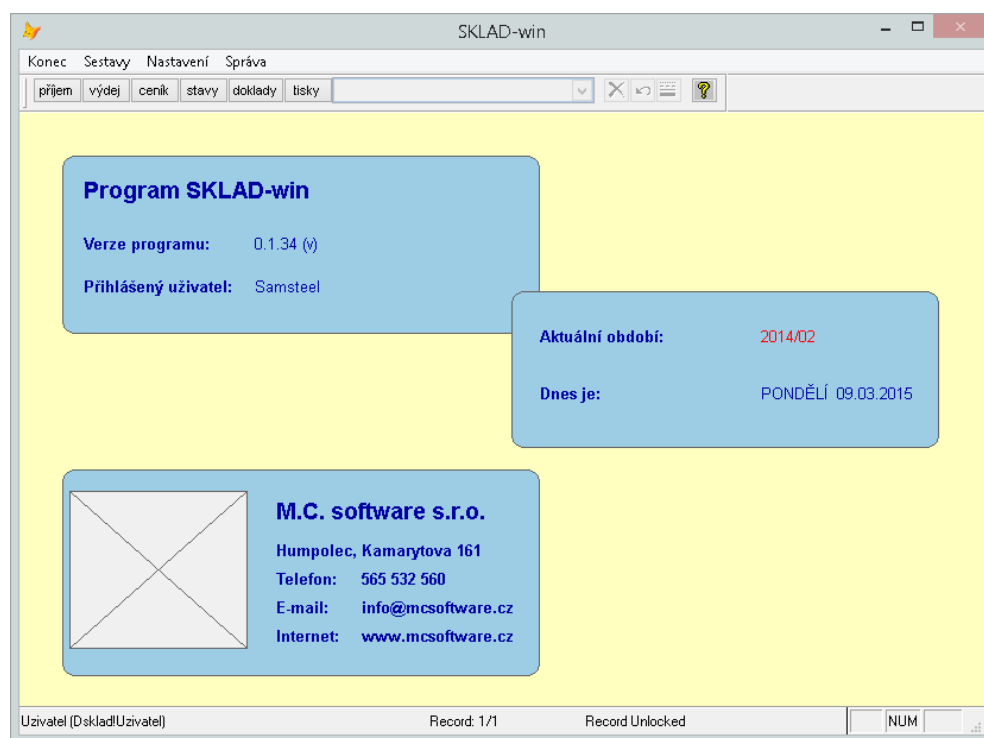
Dle mého názoru je v tato nabídka zbytečná a po přihlášení by se měl spustit okamžitě hlavní program. Žádná z těchto akcí nepatří mezi každodenní provozní akce a pouze to zdržuje uživatele od spuštění programu. Myslím si, že by se tato nabídka měla v upravené verzi objevit zakomponovaná do hlavního programu.

2.2.1.3 Hlavní okno programu

Hlavní menu programu se skládá z „klasické“ výsuvné nabídky a z tlačítek rychlého přístupu, kde jsou nejčastěji používané funkce programu. Pod položkou sestavy nalezneme výběr ze základní nabídky reportů k tisku. Nastavení obsahuje přístup ke všem číselníkům, ze kterých se následně skládají příjemky a výdejky (ceník, adresář

partnerů, střediska, zakázky, sklady a účetní skupiny) a zároveň samotné nastavení programu - údaje o společnosti, nastavení číselných řad pro jednotlivé doklady, výchozí měny se kterými program bude operovat. Obsahuje také přístup na samotný formulář nastavení programu, kde je možné nastavit výchozí parametry jednotlivých formulářů.

Pod menu je možné vidět základní informace o programu, kontaktní údaje na podporu a informace o nastavení období a dnešního data programu. V této oblasti se zároveň spouští všechny formuláře, které si uživatel otevře.



Obrázek 5 - Hlavní okno původní verze programu

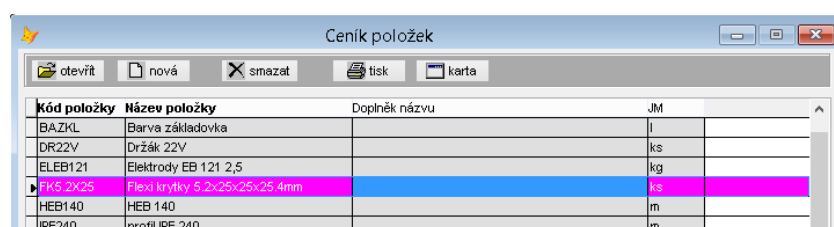
Hlavní okno programu dnes již vypadá velice zastarale – celkově je nutné vzhled programu vylepšit. Velice prakticky působí i horní lišta, kde se nachází nejpoužívanější formuláře programu – příjem, výdej, ceník, stavy, doklady, tisky a tak jsou uživatelé rychle přístupné. Na druhou stranu v moderních aplikacích se dnes setkáváme s možností úpravy těchto „rychlých odkazů“ na základě uživatelského nastavení. Tato možnost v programu chybí.

Rozložení hlavní nabídky je správně logicky uspořádané – pod položkou sestavy může uživatel najít všechny reporty a informační formuláře, v položce nastavení číselníky

a další nastavení programu a v sekci správa nabídku s netypickými formuláři. Orientace v této nabídce je velice jednoduchá a intuitivní.

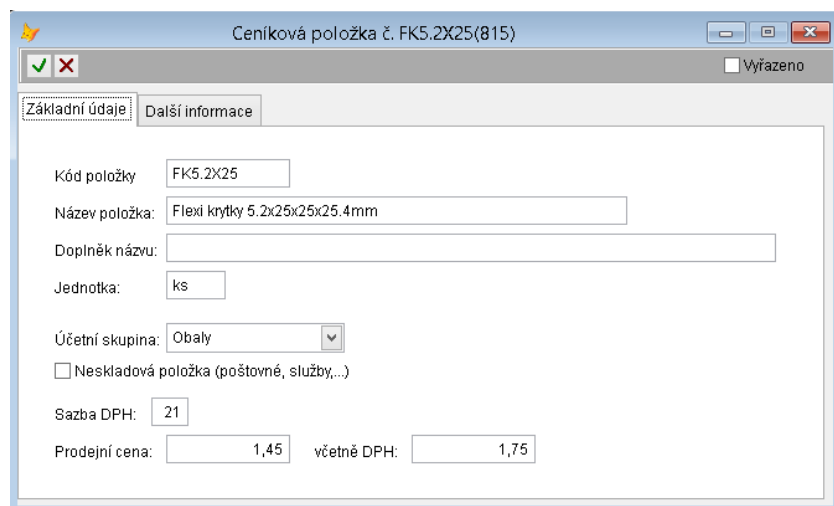
2.2.1.4 Příklad číselníku – ceník

Po otevření nabídky nastavení – ceník se zobrazí formulář s názvem ceník položek. Tento formulář se skládá z horní nabídky, kde uživatel může zobrazit detail položky v číselníku (a upravit je), přidat novou položku do číselníku, odstranit vybranou položku v číselníku, vytisknout celý číselník a v případě tohoto číselníku tlačítkem karta, zobrazit kartu se všemi pohyby vybraného zboží. Hlavní částí formuláře s číselníkem je DataGrid se všemi položkami, které se v daném číselníku nachází. V tomto Gridu je možné řadit a filtrovat jednotlivé položky pomocí naprogramovaných formulářů. Podle tohoto modelu jsou vytvořeny všechny číselníky v programu.



Kód položky	Název položky	Doplňk názvu	JM
BAZKL	Barva základovka		l
DR22V	Držák 22V		ks
ELEB121	Elektrody EB 121 2,5		kg
FK5.2X25	Flexi krytky 5.2x25x25x25.4mm		ks
HEB140	HEB 140		m
IPE240	profil IPE 240		m

Obrázek 6 - Číselník z původní verze programu - zavřený



Ceníková položka č. FK5.2X25(815)

☒ Znovu načíst ☐ Vytvářeno

Základní údaje Další informace

Kód položky: FK5.2X25

Název položky: Flexi krytky 5.2x25x25x25.4mm

Doplňk názvu:

Jednotka: ks

Účetní skupina: Obaly

☐ Neskladová položka (poštovné, služby,...)

Sazba DPH: 21

Prodejní cena: 1,45 včetně DPH: 1,75

Obrázek 7 - Editace číselníku v původní verzi programu

Práce s tímto typem formuláře je jednoduchá a intuitivní, ale přijde mi zbytečné, že na většinu akcí otevírá nový formulář. Zabudováním některé funkcionality přímo do hlavního formuláře by se podle mého názoru zvýšila přehlednost práce v programu.

2.2.1.5 Příjem a výdej zboží

Nejčastěji používanými okny celého programu jsou bezesporu příjem a výdej zboží.

Program podporuje následující druhy příjmu zboží:

- nákup od cizích,
- z jiného skladu (vytvoří zároveň 2 doklady - příjemku do cílového skladu a výdejku ze zdrojového skladu),
- z výroby,
- inventurní přebytek,
- jiný příjem,
- z jiného střediska

a následující druhy výdeje zboží:

- prodej cizím,
- do jiného skladu (vytvoří zároveň 2 doklady - příjemku do cílového skladu a výdejku ze zdrojového skladu),
- spotřeba,
- režijní materiál,
- vzorky,
- manka,
- škody,
- jiný výdej,
- do jiného střediska.

Na základě nastavení druhu příjmu/výdeje se přizpůsobuje celý formulář v oblasti zadávání dodavatele/odběratele.

Uživatel v prvním kroku vybere druh pohybu a zvolí dodavatele/odběratele. Následně vybere sklad příjmu/výdeje. V dalším kroku specifikuje položky dokladu – druh a množství zboží. Po vybrání druhu zboží a skladu příjmu/výdeje se zobrazí, kolik kusů zboží je momentálně uskladněno a průměrná skladová cena tohoto zboží. V případě příjmu musí uživatel specifikovat i částku, za kterou je zboží přijímáno. Po stisknutí tlačítka „potvrdit“ se položka uloží do DataGridu pod formulářem. Tímto způsobem zadá uživatel všechno zboží, které chce evidovat na dané příjemce/výdejce. Po uložení dokladu nabídne program uživateli jeho vytisknutí.

Nová příjemka

uložit konec tisk import Částka: 7 500,00

Hlavička dokladu Položky dokladu

Číslo dokladu: 0
Datum příjmu: 28.02.2014
Sklad - příjemce: Sklad 1
Druh příjmu: Nákup od cizích
Dodací list: Faktura: Zakázka: ---
Poznámka:

Dodavatel:
Kód: 0 IČO:
Název:
Ulice, č.p.:
PSČ, město:

Kód položky	Název položky	Množství	JM	Cena/JM	Částka	Doplňk názvu položky
KH14-RAL	KH 14-RAL 5019	5,00	l	1 500,000	7 500,000	

Obrázek 8 - Příjemka v původní verzi programu – hlavička dokladu

Nová příjemka

uložit konec tisk import Částka: 7 500,00

Hlavička dokladu Položky dokladu

Kód: KTM JM: KS
Název: Kamnářský tmel
Množství: 0,00
Cena: 0,00 Částka: 0,00
Stav: 0,00
Cena: 0,00
potvrdit

Kód položky	Název položky	Množství	JM	Cena/JM	Částka	Doplňk názvu položky
KH14-RAL	KH 14-RAL 5019	5,00	l	1 500,000	7 500,000	

Obrázek 9 - Příjemka v původní verzi programu - položky dokladu

Zboží skladový program vydává za skladovou cenu, kterou vypočítá jako vážený průměr cen, za které bylo zboží uskladněno. Další skladovací metody program bohužel nepodporuje.

2.2.1.6 Sestavy a reporty

Jednou z nejdůležitějších funkcí každého informačního systému je podávat aktuální, pravdivé a přehledné informace. K tomuto účelu slouží reporty. V základní verzi programu je nabídka následujících modifikovatelných tiskových sestav:

- součty po skladech,
- součty podle materiálu,
- materiál / sklady,
- celková bilance,
- bilance po skladech,
- bilance po položkách,
- bilance - pouze rozdíly,
- bilance - po účtech,
- inventurní soupis,
- přehled pohybů,
- výdeje na střediska.

SAMSTEEL s.r.o.		Celkový stav ve skladech		Tisk: 11.01.2015 20:08:14
		Třídění: Zkratka vzestupně		Strana: 1
Sklad: Sklad 1 & Skup.: VŠE & Kód zb.: VŠE & Názzb.: VŠE & K. datu: 11.01.2015				
Sklad	Název skladu	Počet položek	Částka	
1	Sklad 1	32	276 744,027	
Celkový stav:		32	276 744,027	

Obrázek 10 - Report v původní verzi programu

Na tomto reportu můžeme vidět celkovou hodnotu zboží v jednotlivých skladech (v této testovací databázi je pouze jeden sklad „Sklad 1“) a celkovou hodnotu zboží ve všech skladech. Hlavička reportu má ve všech reportech generovaných tímto programem stejné rozložení. Nabídka reportů v tomto programu je velice obsáhlá a většina uživatelů zde jistě najde všechny informace, které potřebují.

Co by ale společnost mohla vylepšit, je forma těchto reportů. Grafické znázornění by jistě bylo v některých případech přehlednější, než tabulkový výpis. Za zvážení by také stálo vytvoření modulu pro uživatelské reporty, který by pak společnost mohla využít i v dalších programech, které nabízí.

2.2.1.7 Práce v programu

V této části práce se budu zabývat rychlostí a efektivností práce v programu. [19]

Pro tuto příkladovou studii budeme pracovat se scénářem, kde má pracovník za úkol vytvořit příjemku a upravit položku v číselníku dodavatelů.

Spuštění aplikace sklad, zadání autorizačních údajů (2 akce), stisknutí tlačítka přihlásit, stisknutí tlačítka spustit program, stisknutí tlačítka příjem, výběr skladu příjemce, výběr druhu příjmu (2 akce), výběr dodavatele (3 akce), přepnutí na kartu položka dokladu, výběr zboží (3 akce), zadání množství, zadání částky, stisknutí tlačítka potvrdit, stisknutí tlačítka hlavička dokladu, kontrola zadaných údajů, stisknutí tlačítka uložit, schválit uložení, zavřít okno pro tisk dokladu, zavřít příjemku, stisk tlačítka v menu nastavení, stisk položky adresář, otevření formuláře pro editaci partnera, zobrazení karty další informace, zadání nové hodnoty, stisknutí tlačítka uložit, potvrdit zadané změny, zavřít adresář partnerů, stisknutí tlačítka konec, stisknutí tlačítka konec programu

Tento jednoduchý scénář vyžadoval od uživatele celkem 35 akcí.

2.2.2 Databázová vrstva programu SKLAD

V této části práce se budu zabývat původní databází programu.

2.2.2.1 Přehled současného stavu

Databáze, se kterou pracuje skladový program, se skládá z 19 tabulek. Tyto tabulky nejsou provázané relačními vazbami.

Afm – adresář firem, které jsou následně využívány jako dodavatelé/odběratelé.

Ask – číselník skladů, se kterými program pracuje.

Cenik – v této tabulce je obsažen seznam zboží, se kterým program pracuje.

Cenik_U – nastavení účtů pro jednotlivé účetní skupiny, které nesouvisí přímo se skladovým programem, ale je to nezbytné pro spolupráci s dalšími programy, které společnost nabízí.

Crady – jak již napovídá název, tato tabulka obsahuje seznam číselných řad, pro jednotlivé doklady. Obsahuje číslo posledního dokladu, přírůstek, prefix a afix.

Dodak – tabulka připravená pro budoucí rozšíření v podobě tisku faktur a dodacích listů

Firma – základní informace o společnosti, která program provozuje. Z této tabulky se čerpají údaje např. do hlaviček reportů nebo na příjemky/výdejky.

Meny – tabulka připravená pro práci s cizími měnami.

Nast – obsahuje tabulku s nastavením pro jednotlivé uživatele. Jedná se zejména o uložení výchozích parametrů pro jednotlivé formuláře a další uživatelské preference.

Nastimp – tabulka sloužila jako „pomocná tabulka“ při migraci ze starší verze dosového programu.

Pohyb_p – každá položka na každé příjemce či výdejce. Toto je nejdůležitější tabulka v celém programu, na jejím základě se počítají stavy zboží na jednotlivých skladech. Na základě spojení této tabulky a tabulky pohyb můžeme jednoduše sestavit select, který vypíše doklad nebo karty pohybů zboží.

Pohyb – v této tabulce jsou uloženy hlavičky dokladů.

Stav – v této tabulce je uložen součet pohybů, které byly uskutečněny v aktuálně otevřeném období. Při měsíční uzávěrce se zkopírují do tabulky Stav_o.

Stav_o – uložený stav na všech skladech každého zboží na konci každého období. Tabulka slouží jako kontrolní součet a je možné na základě ní dohledat případné nesrovnalosti. Také razantně snižuje výpočetní náročnost základních operací, kde se nemusí počítat se všemi záznamy v tabulce pohyb (kde mohou být po delší době běhu programu milióny záznamů), ale pouze s těmi z posledního období a stavem z posledního období. Zároveň program podle této tabulky určuje aktuálně otevřené období.

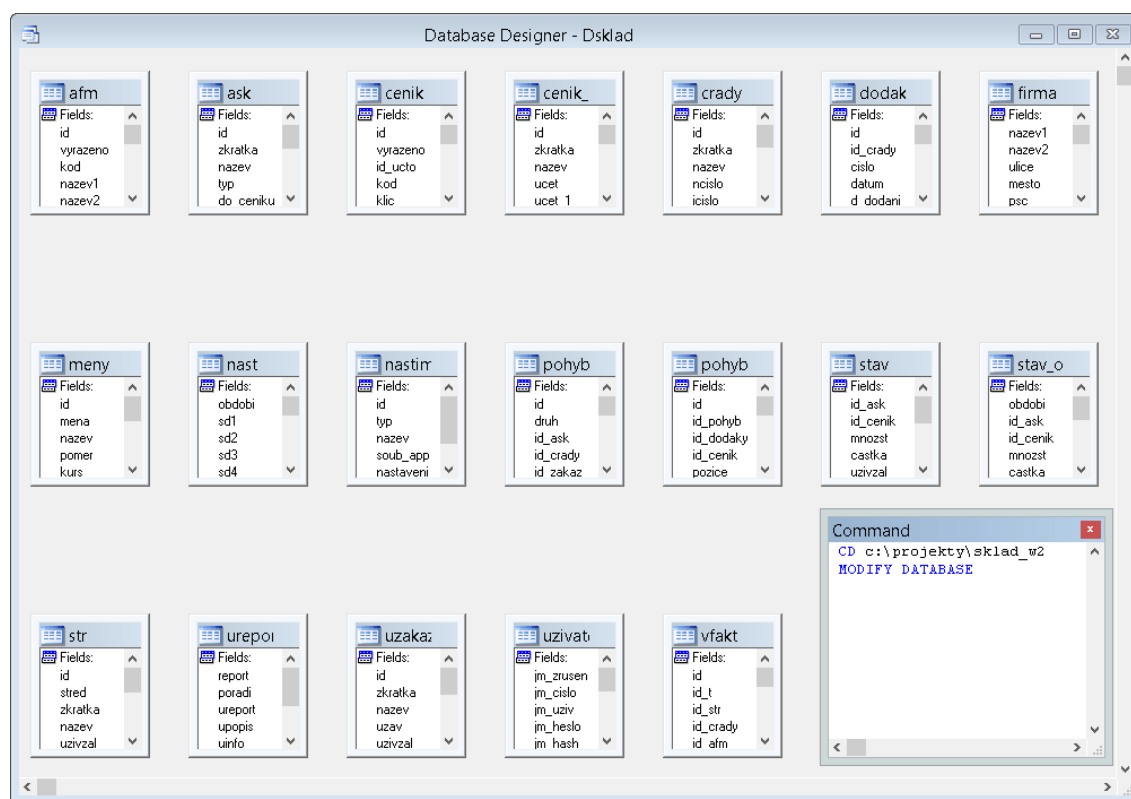
Str – číselník středisek, na která se mohou přiřazovat pohyby.

Ureport – tabulka slouží k uložení nastavení uživatelských reportů, tato funkce však nebyla nikdy plně implementována.

Uzakaz – podobně jako střediska, lze jednotlivé doklady přiřadit k zakázkám. V této tabulce je uložený seznam zakázek.

Uzivatel – tabulka uživatelů s uloženým heslem v uzavřené formě pomocí MD5 hashe. Obsahuje základní informace o uživatelích, včetně oprávnění.

Vfakt – tabulka, která je připravena pro ukládání vystavených faktur. Tato funkcionalita zatím nebyla dopracována.



Obrázek 11 - Databázové schéma v původní verzi programu

2.2.2.2 Hlavní problémy současné databázové vrstvy

Chybí relační vazby mezi jednotlivými tabulkami – skladový program společnost nabízí již od svých počátků. První verze byly navrženy v době, kdy se o relačních vazbách nepřemýšlelo jako o nutném standardu pro podobné aplikace. Tento program bohužel není pro společnost natolik důležitý a tak se stalo, že databázová struktura

nebyla od této verze aktualizována jinak, než přidáním atributů na základě uživatelského požadavku.

Nadbytečnost tabulky stav – v současné době nevidím důvod pro tuto relaci, která sloužila převážně jako kontrolní součet.

Chybějící SŘBD vrstva – jedním z hlavních problémů databázové vrstvy je absence SŘBD vrstvy. Vzhledem k legislativě spojené s účetními doklady (563/1991 Sb.) o úschově dokladů, je nutné příjemky a výdejky archivovat po dobu pěti let. Dojde-li k úpravě položky v libovolném číselníku, není nikdo schopen zobrazit (případně vytisknout) původní verzi dokladu. Tento zásadní nedostatek by bylo možné kompletně vyřešit pomocí triggerů. [21]

2.2.2.3 Kladné stránky databázové vrstvy

Atributy – lze říci, že atributy všech relací jsou vyladěny dlouholetým provozem programu a zpětnou vazbou mnoha uživatelů.

2.2.3 Okolí informačního systému

Okolí informačního systému budu analyzovat ze tří hlavních hledisek – legislativa, použité technologie a spolupráce s dalšími informačními systémy.

2.2.3.1 Legislativa

Příjemka a výdejka – v souladu s ustanovením §11 zákona č. 563/1991 Sb., o účetnictví, musí každý účetní doklad obsahovat: „

- a) Označení účetního dokladu,*
- b) obsah účetního případu a jeho účastníky,*
- c) peněžní částku nebo informaci o ceně za měrnou jednotku a vyjádření množství,*
- d) okamžik vyhotovení účetního dokladu,*
- e) okamžik uskutečnění účetního případu, není-li shodný s okamžikem podle písmene d),*
- f) podpisový záznam podle § 33a odst. 4 osoby odpovědné za účetní případ a podpisový záznam osoby odpovědné za jeho zaúčtování.“ [20]*

Příjemky a výdejky musí být uchovány po dobu pěti let. [21]

2.2.3.2 Použité technologie

Vývojové prostředí Visual Fox Pro 9.0., které bylo použito pro vytvoření poslední verze programu, dostalo poslední update v říjnu 2007 – jedná se tedy o téměř **8 let neaktualizované vývojové prostředí**. Aplikace je překvapivě plně funkční na Windows 8, ale zda tomu tak bude i na připravovaných Windows 10 dnes nikdo ve společnosti s jistotou říci nemůže. Jediným kladným aspektem použitých technologií jsou velice nízké hardwarové nároky na práci s tímto programem. [22]

2.2.3.3 Spolupráce s dalšími informačními systémy

Program komunikuje s dalším softwarem pomocí exportů do XLS a DBF souborů.

3 NÁVRH VLASTNÍHO ŘEŠENÍ

V první části této kapitoly se budu zabývat obecnými postupy a výběrem vývojového prostředí, ve kterém by společnost měla optimalizovanou verzi programu vyvíjet. V druhé části představím dva návrhy na řešení problémů aplikační vrstvy. Ve třetí části představím optimalizovaný databázový model skladového programu. Celou kapitolu uzavřu ekonomickým zhodnocením celého projektu.

3.1 Vývojové prostředí a architektura

Hlavními východisky pro tuto část práce jsou znalosti a zkušenosti zaměstnanců společnosti M.C. software s.r.o. a licence nástrojů, které společnost vlastní.

3.1.1 Vývojové prostředí a nástroje

Společnost M.C. software v současné době vlastní licenční balík Microsoft Action Pack, který obsahuje nejnovější verze nástrojů Microsoft Visual Studio a Microsoft SQL Server. Zaměstnanci mají s tímto prostředím zkušenosti a jedná se o kvalitní vývojové prostředí - společnost nemá dle mého názoru pořizovat jiné vývojářské nástroje.

3.1.2 Architektura informačního systému

Vhodné řešení je podle mého názoru dvouvrstvá architektura s tlustým přístupem. Společnost má s touto architekturou nejvíce zkušeností. Zároveň to umožní přístup k aktuálním informacím o stavech zboží všem pracovníkům ve společnosti bez většího zatížení serveru. Někteří klienti mohou ocenit i možnost spuštění databázového serveru v režimu localhost, kde není vyžadován fyzický databázový server.

3.2 Aplikační vrstva

Společnosti se nabízejí dvě možné cesty, kterými se může vydat v rámci přepracování aplikační vrstvy programu.

3.2.1 Konzervativní přístup

Společnost má velké procento dlouhodobých zákazníků, kteří by pravděpodobně ocenili co nejmenší změny v uživatelském rozhraní. V programu se za dlouhou dobu jeho používání dobře orientují a větší zásahy do uživatelského rozhraní by jim práci pouze komplikovaly. Na druhou stranu, by tato verze návrhu vyřešila problémy se zastaralými technologiemi.

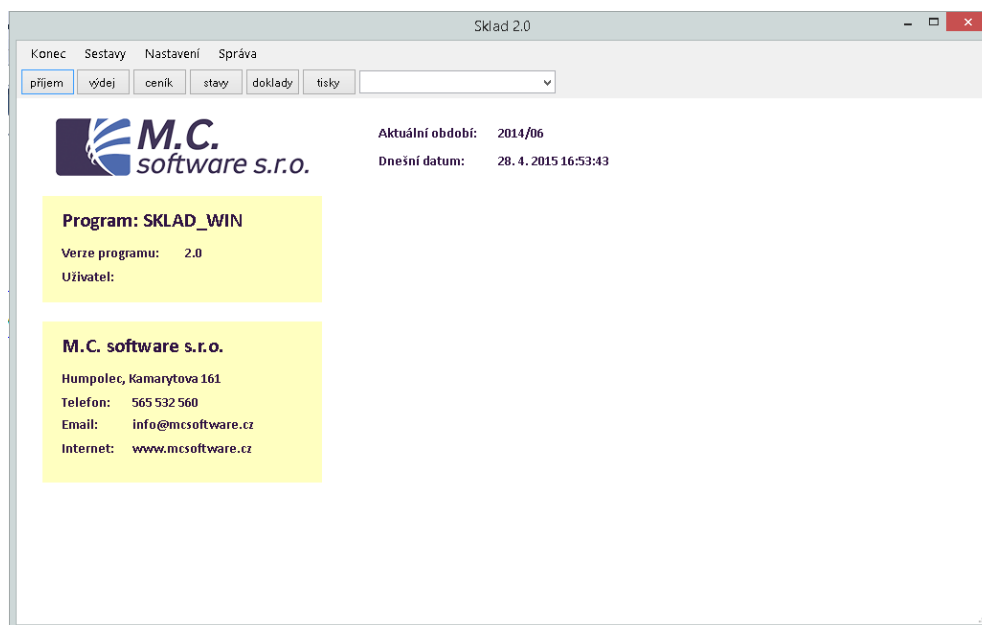
Uživatelské rozhraní programu by tak po zvolení tohoto návrhu dostalo pouze minimálních změn. Rozložení ovládacích prvků na jednotlivých formulářích je téměř identické jako v původní aplikaci. Jediné výraznější úpravy v tomto návrhu jsou odstranění přežitků ze starších verzí programu. Návrh je sestavený v programu Microsoft Visual Studio 2013 s využitím .NET Framework 4.5.

3.2.1.1 Základní nabídka programu



Obrázek 12 - Návrh základní nabídky programu - .NET Frameworku 4.5

3.2.1.2 Hlavní okno programu



Obrázek 13 - Návrh hlavního okna programu - .NET Framework 4.5

3.2.1.3 Příklad číselníku

Ceník materiálů			
Otevřít	Přidat	Karta	Filtrovat
Tisk			Vyřadit
Kód	Položka	JM	Doplněk názvu
ACS01	Míče	m	pěkný
ACS02	Kopačky	páry	Ne tolik pěkný
ACS03	Štrupny	ks	Ne tolik pěkný
ACS01	Míče	m	pěkný
ACS02	Kopačky	páry	Ne tolik pěkný
ACS03	Štrupny	ks	Ne tolik pěkný

Obrázek 14 - Návrh číselníku - .NET Framework 4.5

Ceník - upravit položku	
<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Vyřadit	
Základní údaje	Rozšířené údaje
Kód položky:	ACS03
Název položky:	Štrupny
Doplněk názvu:	Nepovinný popis
Jednotka:	ks
Účetní skupina:	Materiál
<input type="checkbox"/> Neschlaková položka (poštovné, služby...)	
Sazba DPH:	
Prodejní cena:	Včetně DPH:

Obrázek 15 - Návrh na editaci číselníku - .NET Framework 4.5

3.2.1.4 Příjemky a výdejky

Obrázek 16 - Návrh příjemky - .NET Framework 4.5

3.2.1.5 Reporty

Obrázek 17- Návrh reportu - .NET Framework 4.5

Jak lze vidět na obrázcích č. 12-17, ovládací prvky .NET Frameworku dodaly programu modernější vzhled, ale z pohledu uživatele program zůstal téměř nezměněný. Tento model řešení by ušetřil i náklady na případné přeškolení současných uživatelů.

3.2.2 Moderní přístup

Druhou cestou, kterou se společnost může vydat, je cesta výrazné modernizace celého uživatelského rozhraní. Společnost by tím pravděpodobně nepotěšila konzervativní část zákazníků, na druhou stranu by ji moderní vzhled programu mohl pomoci při oslovení nových zákazníků. K vytvoření toho návrhu jsem využil doplňků třetí strany – rozšíření pro Windows Forms aplikace, které nabízí společnost Developer Express Inc. Ačkoliv je toto rozhodnutí na jednatel společnosti, myslím si, že přínosy, které souvisí s doplňkem DevExpress rozhodně převažují náklady na pořízení tohoto nástroje. Mezi jeho hlavní výhody patří:

- 130 moderně vypadajících inteligentních ovládacích prvků,
- inteligentní nástroje na optimalizaci kódu,
- lepší výkon ovládacích prvků,
- lepší úroveň reportů (včetně podpory uživatelských reportů),
- jednodušší a rychlejší implementace,
- podpora dotykového rozhraní,
- česká lokalizace všech ovládacích prvků,
- kvalitní dokumentace společně s ukázkovými videi na řešení běžných úkolů,
- možnost využití tohoto rozšíření i u dalších aplikací, které společnost nabízí.

[23]

Společnosti M.C. software s.r.o. doporučuji investovat do doplňku DevExpress a vydat se touto cestou přepracování programu.

3.2.2.1 Základní nabídka programu

Zrušení základní nabídky – spouštění základní nabídky je z mého pohledu zbytečné zdržování uživatelů, kteří chtějí rychle spustit program a vykonat jednoduchý úkon. Nevidím žádný výrazný důvod, proč položky z této nabídky nezačlenit do hlavní navigace programu.

3.2.2.2 Hlavní okno programu

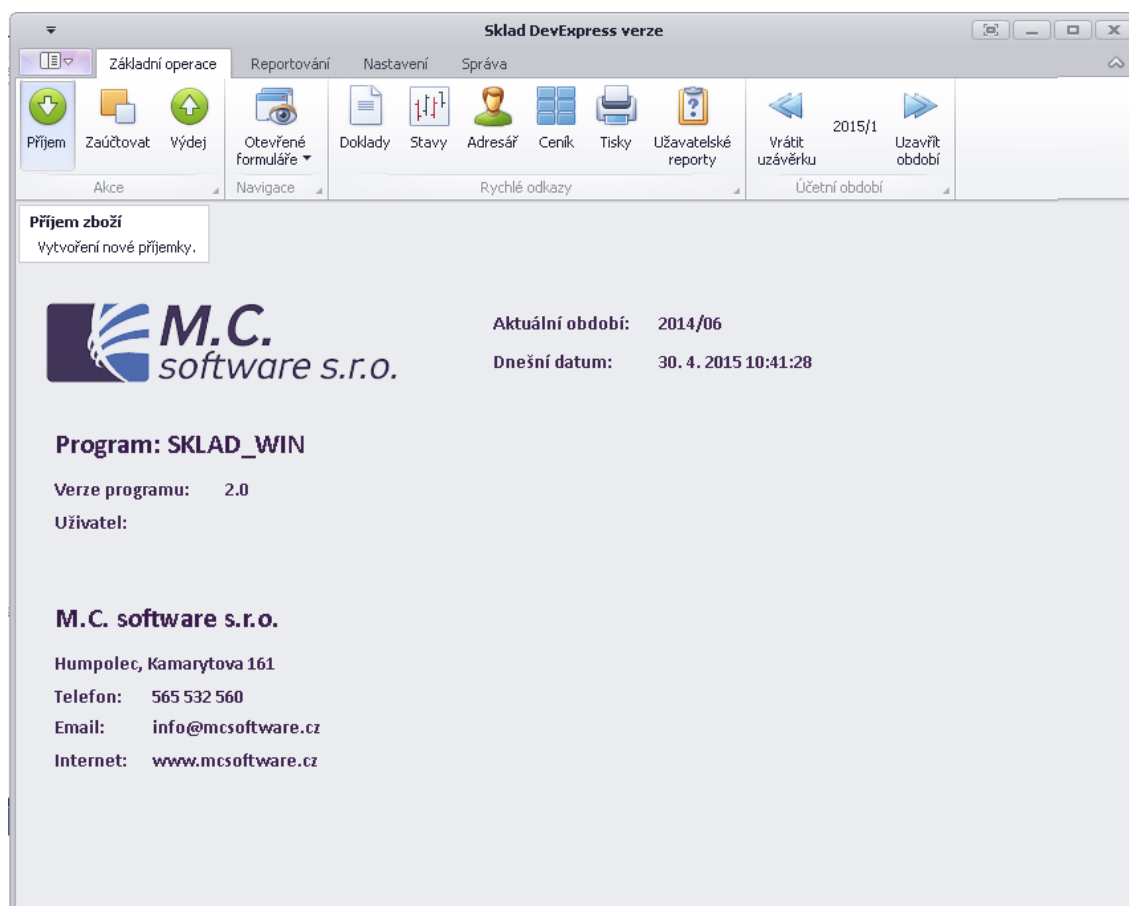
Nové uživatelské rozhraní – další velice výraznou změnou tohoto návrhu je přepracování klasického rozevíracího menu do moderní navigace s využitím ovládacího prvku RibbonControl, který je součástí rozšíření DevExpress. Z důvodu začlenění základní nabídky do tohoto menu se pozměnila i jeho struktura, která je nyní rozdělena do čtyř hlavních kategorií – základní operace, reporty, nastavení a správa programu. Celá tato nabídka je oživena ikonami rovněž z rozšíření DevExpress.

Nabídka základní operace v sobě ukrývá všechny akce, které uživatel tohoto programu využívá při každodenní činnosti nejčastěji – příjem a výdej zboží, skupinu rychlých odkazů. Podobně jako v předchozí verzi zde uživatel najde i výčet aktuálně otevřených formulářů, mezi kterými může přepínat. Nově zde uživatel nalezne i tlačítka na ovládání aktuálně otevřeného účetního období.

V kategorii reporty se uživateli zobrazí seznam všech reportů rozdělených do tří skupin: bilance, stavy a číselníky. Nově má také možnost otevřít nabídku uživatelských reportů, kde může buď vytvořit nový report, nebo upravit jeden z přednastavených reportů.

V položce nastavení se ukrývají všechny číselníky, se kterými program pracuje, a základní nastavení programu – nastavení firmy, nastavení výchozích hodnot v programu, nastavení účetních skupin a editaci rychlých odkazů.

Poslední kategorie je kategorie správa, kam je mimo jiné přesunut i zbytek základní nabídky programu. Uživatel zde najde možnost vytvoření zálohy databáze, možnost obnovy databáze ze zálohy, zobrazení informací o licenci. Pokud má uživatel administrátorské oprávnění, nalezne zde také možnost založení nového uživatele programu a správu oprávnění u současných uživatelů.



Obrázek 18 - Návrh hlavního okna programu - DevExpress

3.2.2.3 Příklad číselníku

Číselníky jsem v této verzi návrhu vytvořil pomocí ovládacího prvku DevExpress DataGridControl. Hlavní přínosy tohoto řešení:

- Snadná implementace – tento ovládací prvek je velice inteligentní a má v sobě zabudované všechny základní funkce, které uživatel bude při práci s DataGridem využívat. Za zmínku stojí plně upravitelné EditFormy pro úpravu a přidávání nových záznamů v DataGridu, integrovaná možnost vyhledávání, seskupování a filtrování dat v DataGridu. Díky těmto funkcím se značně zrychlí případná implementace programu.
- Menší počet otevřených formulářů – díky zabudování editovacích, přidávacích a filtrovacích formulářů přímo do DataGridu se razantně zmenší počet otevřených formulářů, které uživatel využije při běžné práci v programu. S menším počtem otevřených formulářů se zvyšuje přehlednost práce v programu.

Stavy na skladech

Drag a column header here to group by that column

Kod	Název zboží	Množství	Jedn...	Částka	Cena	Sklad
ACS02	Kopačky	4	páry	2080	520	Sklad nad Lipou
ACS03	Štrupny	1	ks	199	199	Sklad nad Lipou
ACS01	Míče	9	m	6300	700	Sklad pod Lipou

Obrázek 19 - Návrh číselníku - DevExpress

Stavy na skladech

Drag a column header here to group by that column

Název zboží	Jednotky	Kod	Množství	Částka	Cena	Sklad
Kopačky	páry	ACS02	4	2080	520	Sklad nad Lipou
Štrupny	ks	ACS03	1	199	199	Sklad nad Lipou
Míče	m	ACS01	9	6300	700	Sklad pod Lipou

Název zboží: Jednotky: Kod:

Množství: Částka: Cena:

Sklad:

Obrázek 20 - Návrh editace číselníku - DevExpress

Stavy na skladech

Drag a column header here to group by that column

Název zboží	Jednotky	Kod	Množství	Částka	Cena	Sklad
Kopačky	páry	ACS02	4	2080	520	Sklad hád Lipou
Štrupny	ks	ACS03	1	199	199	Sklad hád Lipou

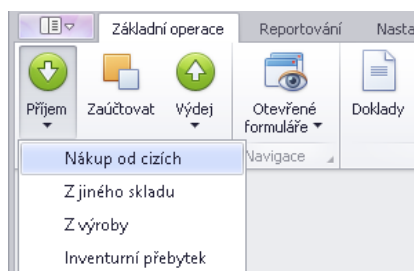
Obrázek 21- Vyhledávání v číselníku - DevExpress

Kod	Název zboží	Množství	Jednotky	Částka	Cena
▼ Sklad: Sklad nad Lípou					
ACS02	Kopačky	4	páry	2080	520
ACS03	Štrupny	1	ks	199	199
► ▼ Sklad: Sklad pod Lípou					
ACS01	Miče	9	m	6300	700

Obrázek 22 - Návrh shlukování a filtrování v číselníku - DevExpress

3.2.2.4 Příjemka a výdejka

V tomto návrhu jsem výrazně přepracoval i nejčastěji používané formuláře – formuláře pro příjem a výdej zboží. Samotné tlačítko, kterým spustí uživatel vytvoření příjímky/výdejky je umístěno velice podobně jako v původní programu, ale v mém návrhu se po jeho stisknutí zobrazí nabídka s výběrem druhu příjmu/výdeje. Uživateli se tak zobrazí formulář v konečné podobě a nemění se na základě volby uvnitř formuláře.



Obrázek 23 - Návrh příjímky - výběr druh dokladů - DevExpress

Podstatné změny jsem udělal i na samotném formuláři příjímky/výdejky. První zásadní úprava je zrušení rozdělení na hlavičku dokladu a položky dokladu – uživatel má v mém návrhu vše přehledně uspořádáno na jedné „stránce“ podobně, jako by tomu bylo v tištěné verzi dokladu. Uživatel tak před sebou pořád vidí, co má přesně v tomto dokladu zadané a nemusí tak neustále přepínat mezi hlavičkou dokladu a položkami dokladu.

Dále jsem provedl několik drobných změn, které by měly umožnit uživateli rychlejší práci v programu. V horní části formuláře má nyní dvě tlačítka – uložit a vytisknout a uložit a zavřít. Uživatel tak nemusí neustále odpovídat na formulář, co chce s daným dokladem udělat (vytisknout/nevytisknout). Dále jsem využil ovládací prvek z rozšíření

DevExpress GridLookUp, který usnadňuje vyhledávání v číselnících, takže uživatel již nemusí pro výběr dodavatele či zboží spouštět další formulář.

Příjemka č. 100001

Uložit a zavřít Uložit a vytisknout Celkem: 7500.00 Kč

Sklad - příjemce: Sklad pod Lípou Dodací list: Číslo dokladu: 100001
 Zakázka: Faktura: Datum příjmu: 18. dubna 2015

Kód: 12345678 DODAVATEL Společnost: Test Dodav s.r.o.
 dodav kopačky, miče 12345678
 PSČ, město: 39601 Humpolec
 Ulice a č.p.: Lipnická 123

Kód: ACS02 ZBOŽÍ Stav: 0
 Název: Kopačky
 Množství: ID KOD NÁZEV ČÁSTKA
 4 ACS01 Miče
 5 ACS02 Kopačky
 6 ACS03 Štrupny
 Částka: x

Drag a column header here to group by that column

KÓD:	NÁZEV POLOŽKY:	MNOŽSTVÍ:	JM:	CENA/CM	ČÁSTKA:	DOPLNĚK NÁZVU:
ACS02	Kopačky	5	páry	1500 Kč	7500 Kč	červené

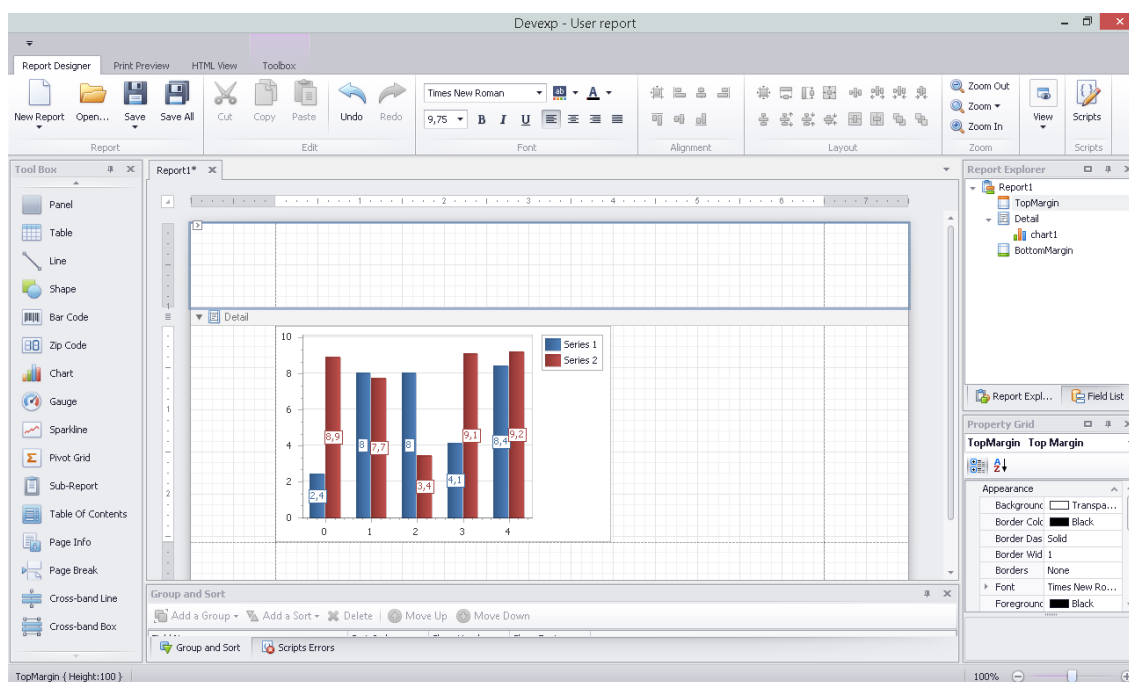
Poznámka:

Obrázek 24 - Návrh příjemky - DevExpress

3.2.2.5 Sestavy a reporty

Největším přínosem přepracování programu pomocí knihoven společnosti DevExpress jsou bezesporu reporty. Oproti základnímu .NET 4.5 Frameworku dostává vývojář širokou škálu nástrojů včetně vybavy pro vytváření grafických reportů nebo plně funkční ovládací prvek pro uživatelské reporty, kde si uživatel bez problémů může upravit přednastavené reporty, nebo i vytvořit zcela nový report.

Prostředí pro uživatelské reporty je velice jednoduše uspořádáno a věřím, že po krátkém zaškolení by v něm byl schopen pracovat i méně zkušený uživatel. Jistě pomůže celková česká lokalizace.



Obrázek 25 - Uživatelské reporty - DevExpress

3.2.2.6 Příklad rutinní akce

Zde budu pracovat se stejným scénářem jako v kapitole 2.2.1.7, tedy že pracovník má za úkol vytvořit příjemku a upravit položku v číselníku adresář. V novém návrhu musí provést následující akce:

Spuštění aplikace, zadání autorizačních údajů (2 akce), stisknutí tlačítka přihlásit, stisknutí tlačítka příjem, výběr druhu příjmu, výběr dodavatele (2 akce), výběr zboží (2 akce), zadání množství, zadání částky, stisknutí tlačítka potvrdit, kontrola zadaných údajů, stisknutí tlačítka uložit, stisk rychlého tlačítka adresář, otevření editace partnera, zadání nové hodnoty, stisknutí tlačítka uložit, zavřít adresář partnerů, stisknutí tlačítka pro zavření programu, potvrzení ukončení programu.

V nově navrhnutém prostředí tento scénář vyžadoval od uživatele pouze 22 akcí místo původních 35 akcí.

3.3 Databázová vrstva

Databázovou vrstvu programu jsem rozdělil do dvou kategorií – návrh relačního modelu a návrh systému řízení báze dat.

3.3.1 Návrh relačního modelu

Na obrázku č. 26 je vidět zjednodušený návrh optimalizovaného databázového modelu. Při vytváření návrhu relačního modelu databáze jsem vycházel z modelu původního programu, který společnost vytvořila. Zachoval jsem původní názvy všech relací a atributů, upravil jsem pouze problémy zmíněné v kapitole 2.2.2.2.

Vytvoření relačních vazeb – v původním databázovém modelu společnosti byly vytvořeny vazby - relace A obsahovala hodnotu primárního klíče relace B, ale nebylo ve vlastnostech těchto atributů specifikováno, že se jedná o cizí klíče. Neplatila tak referenční integrita databáze. V novém návrhu jsem tento problém odstranil.

Odstranění relace stav – jedná se o odvozenou relaci, v návrhu jsem ji odstranil a nahradil pohledem.

Drobné úpravy atributů – převážně odstranění atributů, které již nebyly využívány aktuální verzí programu.

Pro větší přehlednost databázového modelu jsem se rozhodl z tohoto návrhu odstranit tabulky „uzivatel“ a tabulku „nast“, které byly navázané téměř na všechny relace v návrhu („nast“ z důvodu obsahu výchozích hodnot pro všechny číselníky, „uzivatel“ z důvodu záznamů o založení a poslední úpravě každé relace).

3.3.2 Návrh SŘBD

V systému řízení báze dat jsem využil pohledy a spouště.

3.3.2.1 Pohledy

Pohledy je vhodné používat u všech často využívaných složitých databázových selectů. Podle mého názoru nejlepší využití tohoto nástroje je vytvoření pohledu pro každý report, se kterým program pracuje (viz. 2.2.1.6) i pro další často se zobrazující se selecty, jako jsou například karta zboží, stavy zboží a seznam dokladů. Tato separace složitých databázových dotazů z kódu programu by se zjednodušila a zrychlila budoucí aktualizace programu i jeho samotnou implementaci.

3.3.2.2 Triggery

Pomocí triggerů se dá snadno vyřešit problém archivace viz. kapitola 2.2.2.2. Do každého číselníku, který je přímo vázán na daňové doklady, bych přidal atribut archiv s datovým typem bit, o velikosti jedna. Celý program bude následně pracovat pouze se záznamy, kde bude tato hodnota rovna 0. Záznamy, ve kterých bude tato hodnota rovna 1, budou archivní hodnoty, které bude program využívat pouze při zobrazování daňových dokladů. Následně stačí vytvořit jednoduché triggery místo příkazů update a delete nad zmíněnými číselníky. Operace delete se nahradí jednoduchým updatem, který hodnotu atributu archiv změní z hodnoty 0 na hodnotu 1. V případě operace update nastane stejná akce jako místo operace delete, ale následně dojde i k vložení nového záznamu do relace, který bude obsahovat upravené hodnoty a hodnota atributu archiv zde bude rovna 0. Z pohledu uživatele tedy nastane update/delete, ale v databázi zůstanou uložené i původní hodnoty, které umožní zobrazit doklad vždy v původní verzi.

3.4 Nástin modernějšího přístupu k problematice

Při současném trendu v oblasti podnikových informačních systémů, se nabízí otázka, zda je přepracování skladového programu tak, jak si ho společnost představuje, správnou volbou optimalizace nabízených programů. Podle posledních trendů migrují moderní společnosti z funkčně orientovaného modelu na procesně orientovaný model.

Vedení společnosti by se tedy mělo zamyslet, zda chce konzervativně stále nabízet funkčně orientovaný model podnikového informačního systému, nebo zda sloučí některé ze svých programů a nabídne svým zákazníkům i procesně orientovanou verzi svých programů. Takto rozsáhlá změna nabídky společnosti by byla jistě velice

nákladná a i přes fakt, že by to byl pro společnost jasný krok dopředu, nikdo nemůže s jistotou říci, zda by se tento krok společnosti vyplatil.

3.5 Ekonomické zhodnocení práce

V této části nejprve zhodnotím přínosy mé práce pro společnost a odhadnu náklady, které by společnost musela vynaložit na uskutečnění projektu.

3.5.1 Přínosy navrhovaného řešení

Přínosy navrhovaného řešení bez ohledu na vybranou verzi řešení projektu:

- **Dočasné odstranění technologických rizik** – využití moderních vývojových nástrojů společnosti Microsoft se sníží riziko spojené se zastaralými technologiemi, použitými při vývoji.
- **Archivace dokladů** – díky nově navrhnutému databázovému systému se zajistí zobrazování daňových dokladů vždy v původním znění v souladu se zákonem o účetnictví, bez ohledu na další akce uživatele.

V případě zvolení moderního přístupu také:

- **Zrychlení rutinních operací ve skladovém programu** – z výsledků testu rutinních operací v uživatelském prostředí vyšlo najevo, že práce v programu po jeho přepracování může být až o 37% rychlejší. Na zadaný úkon v původním uživatelském prostředí potřeboval uživatel 35 akcí, oproti tomu v nově navrhovaném uživatelském prostředí by potřeboval akcí pouze 22.
- **Lepší informace pro řízení společnosti** – díky možnosti využití grafů a uživatelských reportů by se kvalita reportů výrazně zlepšila. Zákazník tak má lepší přístup k informacím o pohybu zboží a tak i lepší podklady pro rozhodovací proces.
- **Materiál pro zlepšení prezentace produktů společnosti** – s lépe zpracovaným designem výsledné aplikace se společnost může pokusit o prezentace svých produktů a tak získat nové zákazníky.
- **Kvalitní implementační nástroj** – s pořízením doplňku DevExpress by společnost získala vysoce kvalitní nástroj pro vývoj moderních aplikací.

- **Možnost zvýšení ceny nabízeného programu** – po takto rozsáhlém přepracování a zvýšení funkcionality původního programu by společnost měla být schopna prodat licenci skladového programu za cenu vyšší, než tomu bylo u původního programu.

3.5.2 Náklady řešení

Ceny i počty hodin v této práci jsou odhadované, společnost M.C. software s.r.o. nemá přesně danou hodinovou sazbu za interní práce.

3.5.2.1 Práce na projektu

Tabulka 1 - Práce na projektu

Pracovník	Cena za hodinu	Počet hodin	Celková částka:
Čmok Petr	250 Kč	180	45 000 Kč
Jednatel společnosti	850 Kč	10	8 500 Kč
Celkem:	-	-	53 500 Kč

3.5.2.2 Dokončení projektu

Cena implementace obou návrhů bude podle mého názoru velice podobná. V případě konzervativního návrhu společnost ušetří zhruba 20 000 Kč na licenci doplňku DevExpress, ale pravděpodobně implementace projektu bude časově náročnější než s jeho použitím.

Tabulka 2 - Odhadovaná cena implementace projektu

Položka	Cena	Počet hodin	Celková částka
Implementace databáze	700 Kč	16	11 200 Kč
Implementace aplikace	700 Kč	130	91 000 Kč
Celkem:	-	136	102 200 Kč

Tabulka 3 - Odhadovaná cena projektu - DevExpress

Položka	Cena	Počet hodin	Celková částka
Licence	20 000 Kč	-	20 000 Kč
Implementace databáze	700 Kč	16	11 200 Kč
Implementace aplikace	700 Kč	120	84 000 Kč
Celkem:	-	136	115 200 Kč

I přes skutečnost, že odhadovaná cena tohoto řešení je o několik tisíc dražší, společnost by na konci projektu vlastnila lépe prodejný produkt a licence DevExpress.

3.5.2.3 Celkové odhadované náklady

Celkové náklady konzervativního řešení: 155 700 Kč

Celkové náklady moderního řešení: 168 700 Kč (včetně licence DevExpress)

ZÁVĚR

Proces optimalizace informačních systémů je velice složitá záležitost a s tím jsou spojené i vysoké náklady na jeho uskutečnění. S neustálým vývojem v oblasti informačních technologií a často se měnící legislativou spojenou s podnikovými informačními systémy je pro společnosti nutné neustále na tyto změny reagovat. Ale i na této práci je možné vidět, že přínosy tohoto optimalizačního procesu rozhodně nejsou malé. Vedení společnosti by se tedy mělo u každého nabízeného informačního systému zamyslet, zda výnosy spojené s jeho prodejem převýší náklady na jeho optimalizaci.

Dle mého názoru je analyzovaný informační systém natolik zastaralý, že pokud společnost chce tento systém i nadále nabízet, nemá jinou možnost než investovat do vytvoření nové verze programu na základě tohoto návrhu.

Práce mi přinesla mnoho praktických zkušeností z oblasti podnikových informačních systémů, navrhování databázových systémů i implementace v prostředí Microsoft Visual Studio. Společnosti jsem odevzdal společně s návrhem i částečnou implementaci některých částí programu. Věřím, že spolupráce tak byla prospěšná oběma stranám.

SEZNAM POUŽITÝCH ZDROJŮ

- [1] HARDCASTLE, Elizabeth. *Business Information Systems* [online]. 2008 [cit. 2015-01-07]. ISBN 978-87-7681-463-2. Dostupné z: databáze bookboon.com
- [2] BURÝ, Alois. *Teorie systémů a řízení* [online]. Ostrava: Vysoká škola Báňská, 2007 [cit. 2015-01-07]. Dostupný z: <http://homen.vsb.cz/~bur50/TAR07.pdf>
- [3] WebFinance, Inc. What is information system? definition and meaning. *BUSINESS DICTIONARY* [online]. ©2015 [cit. 2015-01-07]. Dostupné z: <http://www.businessdictionary.com/definition/information-system.html>
- [4] TechTarget. ICT (Information And Communications Technology - Or Technologies) Definition. *SearchCIE*. [online]. ©2007-2015 [cit. 2015-01-11]. Dostupné z: <http://searchcio.techtarget.com/definition/ICT-information-and-communications-technology-or-technologies>
- [5] BRUCKNER, Tomáš, Jiří VOŘÍŠEK, Alena BUCHALCEVOVÁ a kolektiv. *Tvorba informačních systémů: Principy, metodiky, architektury*. Praha: Grada Publishing, 2012. ISBN 978-80-247-4153-6.
- [6] SODOMKA, Petr a Hana KLČOVÁ. *Informační systémy v podnikové praxi*. 2. vyd. Brno: Computer Press, 2010. ISBN 978-80-251-2878-7.
- [7] BASL, Josef a Roman BLAŽÍČEK. *Podnikové informační systémy: Podnik v informační společnosti*. 2. vyd. Praha: Grada Publishing, 2008. ISBN 978-80-247-2279-5.
- [8] Fakulta informatiky MU. Životní cyklus informačního systému. *Fakulta informatiky Masarykovy Univerzity* [online]. 2002 [cit. 2015-01-11]. Dostupné z: <http://www.fi.muni.cz/~smid/mis-zivcyk.htm>
- [9] Microsoft. Ovládací prvky Windows Forms podle funkce. *MSDN – Microsoft Developer Network* [online]. 2015 [cit. 2015-02-19]. Dostupné z: [https://msdn.microsoft.com/cs-cz/library/xfak08ea\(v=vs.110\).aspx](https://msdn.microsoft.com/cs-cz/library/xfak08ea(v=vs.110).aspx)

- [10] CORONEL, Carlos, Steven MORRIS a Peter ROB. *Database Systems: Design, Implementation and Management*. 9. vyd. Boston: Cengage Learning, 2009. ISBN 0538748842.
- [11] HERNANDEZ, Michael J. *Návrh databází*. Praha: Grada, 2006. ISBN 80-247-0900-7.
- [12] KROENKE, David, David J AUER a Jakub GONER. *Databáze*. Brno: Computer Press, 2015, ISBN 978-80-251-4352-0.
- [13] Fakulta dopravní ČVUT v Praze. Vztahy a relace, integrita. *ČVUT v Praze: Fakulta Dopravní* [online]. 2007 [cit. 2015-02-19]. Dostupné z: https://it.fd.cvut.cz/k614daps/prednasky/dps_03_teorie.htm
- [14] CONOLLY, Thomas, Carolyn E BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.
- [15] Microsoft. Základy normalizace databáze. *Microsoft: Support* [online]. ©2015 [cit. 2015-05-19]. Dostupné z: <https://support.microsoft.com/en-us/kb/283878/cs>
- [16] Ministerstvo spravedlnosti České republiky. Veřejný rejstřík a Sbírka listin. *Justice.cz* [online]. ©2012-2014 [cit. 2015-05-19]. Dostupné z: <https://or.justice.cz/ias/ui/rejstrik-firma.vysledky?subjektId=66254&typ=UPLNY>
- [17] M.C. software s.r.o. *Programy pro ÚČETNICTVÍ a MZDY, EVIDENCE DOCHÁZKY* [online]. M.C. software s.r.o., ©2013 [cit. 2015-05-19]. Dostupné z: <http://www.mcsoftware.cz/>
- [18] ČMOK, F. *Návrh marketingového mixu*. Brno, 2012. Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská.
- [19] MOLNÁR, Zdeněk. *Efektivnost informačních systémů*. Praha: Grada, 2000. ISBN 80-7169-410-X.
- [20] HAVIT, s.r.o. Zákon o účetnictví - Část II. - Rozsah vedení účetnictví, účetní doklady, účetní zápisy a účetní knihy. *Business.center.cz* [online]. ©1998-2015 [cit.

- 2015-02-19]. Dostupné z:
<http://business.center.cz/business/pravo/zakony/ucto/cast2.aspx>
- [21] HAVIT, s.r.o. Zákon o účetnictví - Část VI. - Úschova účetních záznamů. *Business.center.cz* [online]. ©1998-2015 [cit. 2015-02-19]. Dostupné z:
<http://business.center.cz/business/pravo/zakony/ucto/cast6.aspx>
- [22] Microsoft. Visual FoxPro Home. *MSDN - the microsoft developer network* [online]. ©2015 [cit. 2015-03-14]. Dostupné z: <https://msdn.microsoft.com/en-us/vfoxpro/bb190225.aspx>
- [23] Developer Express Inc. WinForms Controls | DevExpress. *DevExpress* [online]. ©1998-2015 [cit. 2015-05-19]. Dostupné z:
<https://www.devexpress.com/Products/NET/Controls/WinForms/>

SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

IS - informační systém

ICT - informační a komunikační technologie

ERP - enterprise resource planning

CRM - customer relationship management

SCM - supply chain management

MIS - management information systems

SŘBD - systém řízení báze dat

SEZNAM TABULEK, OBRÁZKŮ A GRAFŮ

Seznam obrázků:

Obrázek 1 - Technologické pojetí informačního systému	15
Obrázek 2 - Organizační struktura společnosti M.C. software .s.r.o.	31
Obrázek 3 - Přihlašovací formulář do původní verze programu	33
Obrázek 4 - Základní nabídka původní verze programu	34
Obrázek 5 - Hlavní okno původní verze programu	35
Obrázek 6 - Číselník z původní verze programu - zavřený	36
Obrázek 7 - Editace číselníku v původní verzi programu	36
Obrázek 8 - Příjemka v původní verzi programu – hlavička dokladu	38
Obrázek 9 - Příjemka v původní verzi programu - položky dokladu	38
Obrázek 10 - Report v původní verzi programu	39
Obrázek 11 - Databázové schéma v původní verzi programu	42
Obrázek 12 - Návrh základní nabídky programu - .NET Frameworku 4.5	46
Obrázek 13 - Návrh hlavního okna programu - .NET Framework 4.5.....	46
Obrázek 14 - Návrh číselníku - .NET Framework 4.5.....	47
Obrázek 15 - Návrh na editaci číselníku - .NET Framework 4.5	47
Obrázek 16 - Návrh příjemky - .NET Framework 4.5.....	48
Obrázek 17- Návrh reportu - .NET Framework 4.5	48
Obrázek 18 - Návrh hlavního okna programu - DevExpress.....	51
Obrázek 19 - Návrh číselníku - DevExpress	52
Obrázek 20 - Návrh editace číselníku - DevExpress	52
Obrázek 21- Vyhledávání v číselníku - DevExpress	52

Obrázek 22 - Návrh shlukování a filtrování v číselníku - DevExpress	53
Obrázek 23 - Návrh příjemky - výběr druhu dokladů - DevExpress	53
Obrázek 24 - Návrh příjemky - DevExpress	54
Obrázek 25 - Uživatelské reproty - DevExpress	55
Obrázek 26 - Zjednodušený návrh databázového schéma.....	57

Seznam tabulek:

Tabulka 1 - Práce na projektu	60
Tabulka 2 - Odhadovaná cena implementace projektu.....	60
Tabulka 3 - Odhadovaná cena projektu - DevExpress	61

SEZNAM PŘÍLOH

Bez příloh.